

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2017

Bc. Daniel Měkyna

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Návrh a implementace řídicích algoritmů pro
sekundární odprášení tandemové pece
Design and implementation of control algorithms for
dedusting system of a tandem furnace

2017

Daniel Měkyna

Zadání diplomové práce

Student: **Bc. Daniel Měkyna**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T041 Řídící a informační systémy
Téma: **Návrh a implementace řídicích algoritmů pro sekundární odprášení tandemové pece**
Design and Implementation of Control Algorithms for Dedusting System of a Tandem Furnace

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Popis technologie tandemové pece a význam sekundárního odprášení.
2. Provedení funkční analýzy technologie.
3. Vytvoření HW konfigurace PLC a tvorba definiční listiny.
4. Návrh a implementace řídicích algoritmů do PLC v souladu s Ingeteam standardem.
5. Simulace řízení pomocí navržených algoritmů.
6. Uvedení zařízení do provozu. Otestování kompletního řídicího systému.
7. Vyhodnocení dosažených výsledků a závěr.

Seznam doporučené odborné literatury:

- [1] BERGER, Hans. *Automating with SIMATIC*. 5th edition. Erlangen, Germany: Publicis Publishing, 2013, 284 p. ISBN 978-3895783876.
- [2] BERGER, Hans. *Automating with STEP 7 in LAD and FBD: SIMATIC S7-300/400 programmable controllers*. 4th rev. and enl. ed. Weinheim: Wiley-VCH, 2008, 440 s. ISBN 978-3-89578-297-8.
- [3] BERGER, Hans. *Automating with STEP 7 in STL and SCL: programmable controllers SIMATIC S7-300/400*. 4th rev. ed. Erlangen: Publicis Corporate Pub., 2007, 527 s. ISBN 978-389-5782-954.
- [4] BALÁTEĚ, Jaroslav. *Automatické řízení*. 1. vyd. Praha: BEN - technická literatura, 2003, 663 s. ISBN 80-730-0020-2.
- [5] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. vyd. Praha: BEN – technická literatura, 1999, 223 s. ISBN 80-860-5658-9.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Vilém Srovnal, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Návrh a implementace řídicích algoritmů pro sekundární odprášení tandemové pece, jsem vypracoval samostatně, pod vedením diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce, jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

28.4.2017

.....

V Ostravě dne



.....

Daniel Měkyna

PROHLÁŠENÍ ZÁSTUPCE FIRMY INGETEAM A.S.

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských oborech VŠB-TU Ostrava.

28.4.2017

.....

V Ostravě dne



Ing. Jaroslav Tyšer
(zástupce firmy Ingeteam a.s.)

PODĚKOVÁNÍ

Touto cestou bych rád poděkoval vedoucímu mé diplomové práce panu Ing. Vilému Srovnalovi, Ph.D za přínosné konzultace a rady, které mi během tvorby této práce poskytl.

Také bych chtěl poděkovat panu Ing. Jaroslavu Tyšerovi a Ing. Antonínu Kučerovi za poskytnutí technického zázemí a mnoha praktických rad.

ABSTRAKT

Tato práce se zabývá realizací řídicích algoritmů pro sekundární odprášení tandemové pece společnosti Arcelor Mittal Ostrava a.s. V první části práce je popsán koncept automatizačního systému s programovatelným automatem Siemens Simatic S7-400 včetně distribuované periferie ET200S a komunikačního protokolu PROFIBUS. V druhé části práce je provedena analýza technického řešení použité technologie odprášení a zasazení technologie do konceptu výroby oceli pomocí tandemové pece. Dále je uvedena funkční analýza celkové technologie s respektováním rozličných provozních stavů. V souvislosti s tím je v třetí části práce uvedena HW konfigurace navrženého řídicího systému, tvorba definiční listiny symbolů PLC a I/O listiny technologie. Stěžejní, čtvrtá část práce se pak zabývá tvorbou řídicích algoritmů a jejich implementací do PLC v souladu se standardy společnosti Ingeteam. Jsou zde popsány všechny klíčové programové prvky a struktury tvořící výslednou logiku řízení. S touto částí také souvisí testování navržených algoritmů v simulačním prostředí WinMOD, implementace diagnostických funkcí a ladění programu. Poslední, pátá část se pak zabývá uvedením řídicího systému do provozu a jeho testováním uvnitř rozvaděče. S tím souvisí otestování vstupů a výstupů připojených do PLC a funkčnosti komunikace s vizualizačním panelem. Závěrem je pak provedeno zhodnocení navržených algoritmů vzhledem k testování navrženého řídicího systému.

KLÍČOVÁ SLOVA

Tandemová pec, Sekundární odprášení, PLC, Simatic S7 – 400, Průmyslová automatizace

ABSTRACT

The thesis deals with the implementation of control algorithms for tandem furnace secondary dedusting technology located in Arcelor Mittal Ostrava a.s. In first part, this work presents the concept of automation system based on Siemens Simatic S7-400 programmable controller including distributed periphery ET200S and PROFIBUS communication protocol. Second part shows an analysis of secondary dedusting technology technical solution with regards to concept of steelmaking in tandem furnace. There is also functional analysis of whole technology with regards to different operating states. In this context, the third part of this work shows HW configuration used for control purposes, definition list of PLC symbols and I/O list for controlled technology. Main, fourth part of this work deals with creation of control algorithms and their implementation in to the PLC according to Ingeteam standards. In this part we can also see description of each program element and structures used in whole control software. Testing those algorithms in simulation software WinMOD, implementation of diagnostic functions and program debugging is also related to this part. Last, fifth part shows final commissioning of designed control system and his testing inside electrical cabinet. This process of final commissioning also includes testing of inputs and outputs connected to the PLC and testing of communication with HMI panel. In the end we can find evaluation of designed algorithms with relation to testing of designed control system.

KEYWORDS

Tandem furnace, Secondary dedusting technology, PLC, Simatic S7 – 400, Industrial automation

SEZNAM SYMBOLŮ A ZKRATEK

CPU – Centrální procesorová jednotka

DB – Data Blok, struktura v PLC obsahující data

FB – Funkční blok, programový prvek v PLC pro zapouzdření kódu

OB – Organizační blok, základní programový prvek v PLC se specifickým významem

PLC – Programmable Logic Controller, programovatelný automat

HMI – Human Machine Interface, vizualizační systém

FBD – Function Block Diagram, programovací jazyk funkčních bloků

I / O – Vstupně – výstupní

STEP7 – Programovací prostředí společnosti SIEMENS

IL – Interlocky, blokové podmínky

OBSAH

1	ÚVOD	1
2	Realizace procesní automatizace pomocí PLC Simatic	3
2.1	PLC Siemens Simatic S7-400	3
2.2	Distribuovaná periferie ET200S a PROFIBUS DP	8
2.3	Programovací prostředky PLC Simatic S7-400	10
3	Popis technologie tandemové pece a význam sekundárního odprášení	12
3.1	Tandemová pec	12
3.1.1	Základní charakteristika technologického celku	13
3.1.2	Popis procesu vzhledem k řízení odprášení.....	14
3.2	Sekundární odprášení tandemové pece (BETH elektrofiltr)	15
4	Funkční analýza sekundárního odprášení.....	17
4.1	Příprava najíždění sekundárního odprášení.....	17
4.2	Najíždění sekundárního odprášení	17
4.3	Normální provoz sekundárního odprášení	18
4.4	Doběh sekundárního odprášení	19
5	Popis řízených komponent technologie.....	20
5.1	Vysokonapěťové agregáty Rico-Werk	20
5.2	Ventilátory, Motorové klapy a tlakové senzory	21
6	Návrh řídicích algoritmů pro sekundární odprášení.....	23
6.1	HW konfigurace PLC S7-400 a ET200S	23
6.2	Navržení HW konfigurace ve STEP7	24
6.3	Tvorba definiční listiny a I/O listu	27
6.4	Programová struktura	30
6.4.1	Struktura instančního databloku	30
6.4.2	Implementace interlockových struktur	33
6.4.3	Kontrola vstupů a výstupů (IO Warning).....	35
6.5	Programové prvky pro realizaci řízení	36
6.5.1	Obecné vlastnosti technologických objektů	36
6.5.2	Blok pro měření analogových veličin AnMeas	40

6.5.3	Blok pro měření digitálních veličin DgMeas	43
6.5.4	Blok pro řízení motorových klap DevTwoPos	44
6.5.5	Blok pro řízení oplachových klap VlvTwoCoil	47
6.5.6	Blok pro řízení ventilátorů a čerpadla DrvOnOff.....	49
6.6	Rozdělení řídicích celků technologie	50
6.6.1	Řídicí algoritmy pro celkové odprášení	50
6.6.2	Řídicí algoritmy pro jednotlivá pole	52
6.7	Diagnostika systému v prostředí STEP7	54
6.7.1	Využití chybových OB pro diagnostické účely	55
6.8	Testování navržených algoritmů pomocí simulace technologie	56
6.9	Uvedení do provozu a testování navržených algoritmů řízení	58
7	Závěr	61
	Literatura	62
	Seznam příloh.....	63

SEZNAM OBRÁZKŮ

Obrázek 2.1: Cyklické provádění uživatelského programu u Simatic S7-400.....	4
Obrázek 2.2: Automatizační systém s PLC Simatic S7-400	5
Obrázek 2.3: PLC Simatic S7-400	6
Obrázek 2.4: Komponenty řídicího systému Simatic S7-400	7
Obrázek 2.5: Výsledné PLC S7-400 použité pro řízení technologie.....	8
Obrázek 2.6: Distribuovaná periferie ET200S	9
Obrázek 2.7: Kompletní distribuovaná periferie ET200S použita pro řízení technologie	9
Obrázek 2.8: Data management v programovatelném automatu Simatic	10
Obrázek 2.9: Příklad segmentu kódu v FBD.....	11
Obrázek 2.10: Příklad segmentu kódu v STL	11
Obrázek 3.1: Tandemová pec Arcelor Mittal a.s.....	12
Obrázek 3.2: Schematické znázornění struktury tandemové pece	13
Obrázek 3.3: Schematické zakreslení primárního a sekundárního odprášení ve vizualizaci pro tandemovou pec	15
Obrázek 3.4: Schematické zakreslení primárního (tmavě šedá) a sekundárního (světle šedá) odprášení TP6.....	16
Obrázek 5.1: Průmyslový PC elektrofiltru Rico-werk	21
Obrázek 6.1: HW konfigurace PLC S7-400 ve STEP7.....	25
Obrázek 6.2: Nastavení parametrů PROFIBUS DP master systému	26
Obrázek 6.3: Konfigurace vzdálené periferie ET200S	26
Obrázek 6.4: Výsledná HW konfigurace řídicího systému ve STEP7	27
Obrázek 6.5: Segment I/O listu pro odprášení BETH TP6	28
Obrázek 6.6: Segment definičního listu pro odprášení TP6.....	29
Obrázek 6.7: Definiční listina a STEP7	29
Obrázek 6.8: Statická oblast instančního DB - public.....	30
Obrázek 6.9: Statická oblast instančního DB – HMI interface	31
Obrázek 6.10: Statická oblast instančního DB – Private.....	31
Obrázek 6.11: Statická oblast instančního DB – rezervované oblasti.....	32
Obrázek 6.12: Kompatibilita statické oblasti instančního DB s definiční listinou.....	32
Obrázek 6.13: Komplexní pohled na strukturu řízení uvnitř PLC	33
Obrázek 6.14: Příklad realizace interlocků v HMI systému	34
Obrázek 6.15: Příklad realizace interlocků ve STEP7	34
Obrázek 6.16: Příklad realizace IO Warningů	35
Obrázek 6.17: Nastavení maximálních časů pro koncové polohy motorových klap	37
Obrázek 6.18: Bloky pro diagnostiku sítě Profibus	38
Obrázek 6.19: Segment intelocků s použitím statusových proměnných.....	39
Obrázek 6.20: Příklad nastavení hladin analogových hodnot	41
Obrázek 6.21: Blok pro realizaci analogových měření AnMeas.....	42

Obrázek 6.22: UDT bloky funkčního bloku AnMeas	43
Obrázek 6.23: Blok pro realizaci digitálních měření (DgMeas)	43
Obrázek 6.24: UDT bloky funkčního bloku DgMeas	44
Obrázek 6.25: Blok pro realizaci řízení motorových klap (DevTwoPos)	45
Obrázek 6.26: Výstupy bloku DevTwoPos	46
Obrázek 6.27: UDT bloky funkčního bloku DevTwoPos	46
Obrázek 6.28: Blok pro realizaci řízení oplachových klap (VlvTwoCoil)	47
Obrázek 6.29: Výstupy bloku VlvTwoCoil	48
Obrázek 6.30: UDT bloky funkčního bloku VlvTwoCoil	48
Obrázek 6.31: Blok pro realizaci řízení ventilátorů a čerpadla (DrvOnOff)	49
Obrázek 6.32: UDT bloky funkčního bloku DrvOnOff	50
Obrázek 6.33: Funkční bloky obsahující algoritmy pro řízení odprášení TP6	50
Obrázek 6.34: Programová struktura návrhu řídicích algoritmů	51
Obrázek 6.35: Funkční bloky jednotlivých polí	52
Obrázek 6.36: Diagnostika navrženého systému – diagnostický buffer	54
Obrázek 6.37: Užití chybové OB pro diagnostiku řídicího systému	55
Obrázek 6.38: Simulace řízení výstupních klap	56
Obrázek 6.39: Simulace řízení oplachových klap	56
Obrázek 6.40: Simulace řízení vysokonapětových sekcí elektrofiltru	57
Obrázek 6.41: Simulace řízení vstupních klap	57
Obrázek 6.42: Simulace chodu celkové technologie sekundárního odprášení	58
Obrázek 6.43: Výsledný vnitřek rozvaděče určený pro řízení sekundárního odprášení	59
Obrázek 6.44: Testování součinnosti vizualizačního panelu a PLC	60

SEZNAM TABULEK

Tabulka 5.1: Seznam řízených komponent sekundárního odprášení	20
Tabulka 6.1: Hardwarová konfigurace PLC	23
Tabulka 6.2: Hardwarová konfigurace ET200S – 1. část	24
Tabulka 6.3: Hardwarová konfigurace ET200S – 2. část	24

1 ÚVOD

Návrh a implementace nových algoritmů řízení jako neodmyslitelná a integrální součást kybernetického řízení v období, které můžeme vnímat jako automatizace 21. století. Stanovení robustního řízení schopného neomylně pracovat desítky let je vize, se kterou pracují všichni bez ohledu na systém či aplikaci.

V oblasti automatického řízení tedy lze vypočítávat masivní snahy zajistit maximální efektivitu řízení komplexních technologických procesů bez ohledu na průmyslové odvětví. Lze tedy hovořit o zpracování tisíců proměnných v reálném čase s ohledem na proměnné podmínky, které lze často v reálném provozu nalézt. Kromě předem známých jevů lze taktéž pozorovat jevy hazardní a náhodné, jež samotné značně ovlivňují výsledné prostředí, ve kterém bude muset výsledný řídicí systém pracovat a na jejichž podmínky musí být schopen reagovat. Můžeme tedy říct, že musí být schopen pružně reagovat na značně dynamické podmínky, které jsou mnohdy závislé na teplotě, ročním období nebo jen na aktuálním vytížení dané technologie.

V této práci tedy bude poskytnut komplexní pohled na sekundární odprašování tandemové pece, jak z hlediska konstrukčního a technického řešení, tak z hlediska řízení celé technologie pomocí programovatelného automatu, konkrétně nejvyšší řady Simatic určené pro komplexní průmyslové aplikace S7-400. Jak bude později detailněji rozebráno, celá technologie čítá přes dva tisíce proměnných, které je potřeba implementovat do navržených řídicích algoritmů. Navržení výsledné řídicí aplikace tedy vyžaduje detailní znalost technologie včetně jejích rozličných technologických stavů, ve kterých se technologie může nacházet v závislosti na fázi tandemové pece. Ve světle toho je poté vytvořen samotný řídicí software, jež je členěn do několika logických částí a doprovodných částí sloužících pro obsluhu komunikace s HMI, zpracování analogových a digitálních hodnot apod. Program je pak realizován v jazyce FBD a STL s tím, že každé pole elektrického filtru představuje jeden funkční blok, kdy každý z funkčních bloků obsahuje kompletní algoritmy řízení všech podprvků, které mají souvislost s funkcionalitou daného pole filtru. To značí, že každé pole má implementováno inteligentní řízení všech motorových klap, čerpadel a napájení vysokonapěťových sekcí tak, aby i přes nestandardní chování nebo nefunkčnost některého prvku byl zachován chod technologie. Proto je na místě korektní použití statusů, příkazů do HMI, varování a jednotlivých bitů interlocků řízených prvků v závislosti na konkrétním algoritmu tak, aby výsledný řízený byl přehledný a maximálně efektivní.

Před samotným uvedením do provozu je taktéž potřeba otestovat navržené algoritmy v jednom ze simulačních softwarů, který emuluje chování systému odprašování v reálných podmínkách. Pomocí tohoto nástroje je možné navržené algoritmy otestovat v součinnosti s HMI a odladit tak dílčí části, bez rizika nechtěného chování za běhu technologie. To v praxi značí možnost simulace otevírání motorových klap, dle námi zvolených časových konstant, s možností signalizace odpichu tandemové pece bez splnění podmínek pro spuštění následné sekvence popřípadě flexibilní přecházení mezi procesy, které jinak na sebe navazují, přičemž splnění jednoho je bezpodmínečně nutné pro spuštění

dalšího. To vše otevírá možnosti pro činnosti, které jinak za chodu technologie není možné provést a výsledné uvádění do provozu je značně zjednodušeno.

Závěrem práce je pak diskuze dosažených výsledků při použití navržených algoritmů vzhledem k testování řídicího systému v rozvaděči technologie, popřípadě poznámky ukazující na diferenci mezi teoretickou a praktickou úrovní navrženého řízení a vyvozené důsledky vzhledem k navrženému řídicímu systému.

2 Realizace procesní automatizace pomocí PLC Simatic

Jako nezbytnou součást procesní automatizace už řadu desetiletí vnímáme programovatelné automaty v roli komponenty zajišťující řízení průmyslových celků. Abychom mohli hovořit o řízení průmyslových procesů je potřeba zmínit historický kontext, který umožnil vývoj v této oblasti až do komplexních průmyslových řešení na míru. Jako předchůdce programovatelných automatů můžeme vnímat reléové řídicí systémy, které i v dnešní době můžeme nacházet v průmyslu použity pro řízení určitých technologických procesů. Postupně s příchodem mikroprocesorů a obecně s rozvojem číslicové techniky byla možnost vzniku prvních programovatelných automatů, jejichž integrální částí byl mikroprocesor představující mozek celého řídicího systému.

Dále v této kapitole bude podrobněji popsán programovatelný automat Siemens Simatic S7-400, jelikož je to produktová řada, která bude konkrétně použita pro realizaci řízení sekundárního odprášení. Kromě toho bude pojednáno i o programovacích prostředcích, které slouží k realizaci řídicích algoritmů pro tuto produktovou řadu.

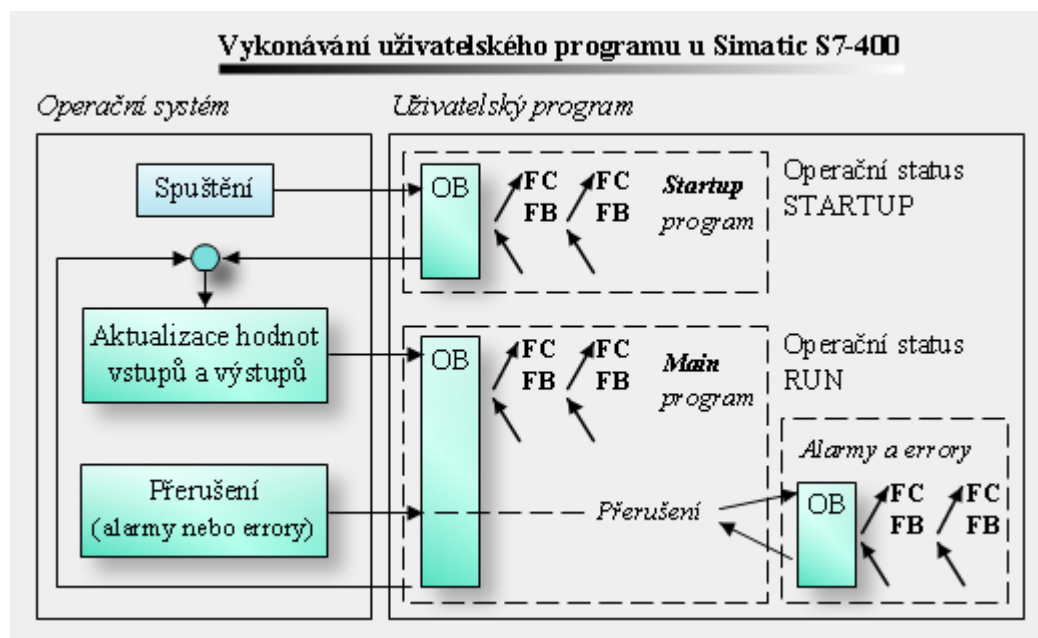
Chceme-li tedy získat detailní představu o celkové koncepci řízení pomocí programovatelných automatů Simatic můžeme využít knih napsaných Hansem Bergerem. Hans Berger je označován jako „otec“ celé produktové řady Simatic S7 a prostřednictvím svých publikací nabízí ucelený a dobře strukturovaný pohled na průmyslovou automatizaci v podání PLC Simatic. Proto i následující kapitoly budou mít nosné myšlenky opírající se o tyto publikace.

2.1 PLC Siemens Simatic S7-400

Jedná se o nejvyšší produktovou řadu programovatelného automatu společnosti Siemens, která je často používána pro řízení velkých technologických celků. Konkrétně se jedná o modulární programovaný automat, tvořen kromě CPU i komunikačními a technologickými moduly přizpůsobitelnými dané aplikaci.

Pokud vezmeme v úvahu zpracování programu, je zde možno pozorovat klasické cyklické provádění uživatelského programu, které je typické pro všechny programovatelné automaty. Tato idea je prezentována na obrázku 2.1, který popisuje logickou sekvenci událostí vedoucí na vykonání uživatelského programu. Po připojení nebo zapnutí zdroje PLC nejprve ověřuje konzistentnost definované hardwarové konfigurace a parametry jednotlivých modulů. Následně je vykonán (pokud je uživatelem naprogramován) startup program (OB 100, 101 a 102), který může zajišťovat inicializaci modulů nebo nastavení určitých inicializačních parametrů uvnitř uživatelského programu. Po proběhnutí této inicializační rutiny dochází k aktualizaci hodnot vstupů a výstupů a je volán hlavní uživatelský program OB1 a všechny ostatní definované funkce a funkční bloky. Po vykonání hlavního programu jsou modifikovány fyzické vstupy a výstupy PLC a programový cyklus se opakuje. Je zde možno podotknout, že cyklické vykonávání uživatelského programu je přerušeno, pokud vznikne hardwarové nebo cyklické přerušení. Hardwarové přerušení pak může být například požadavek od

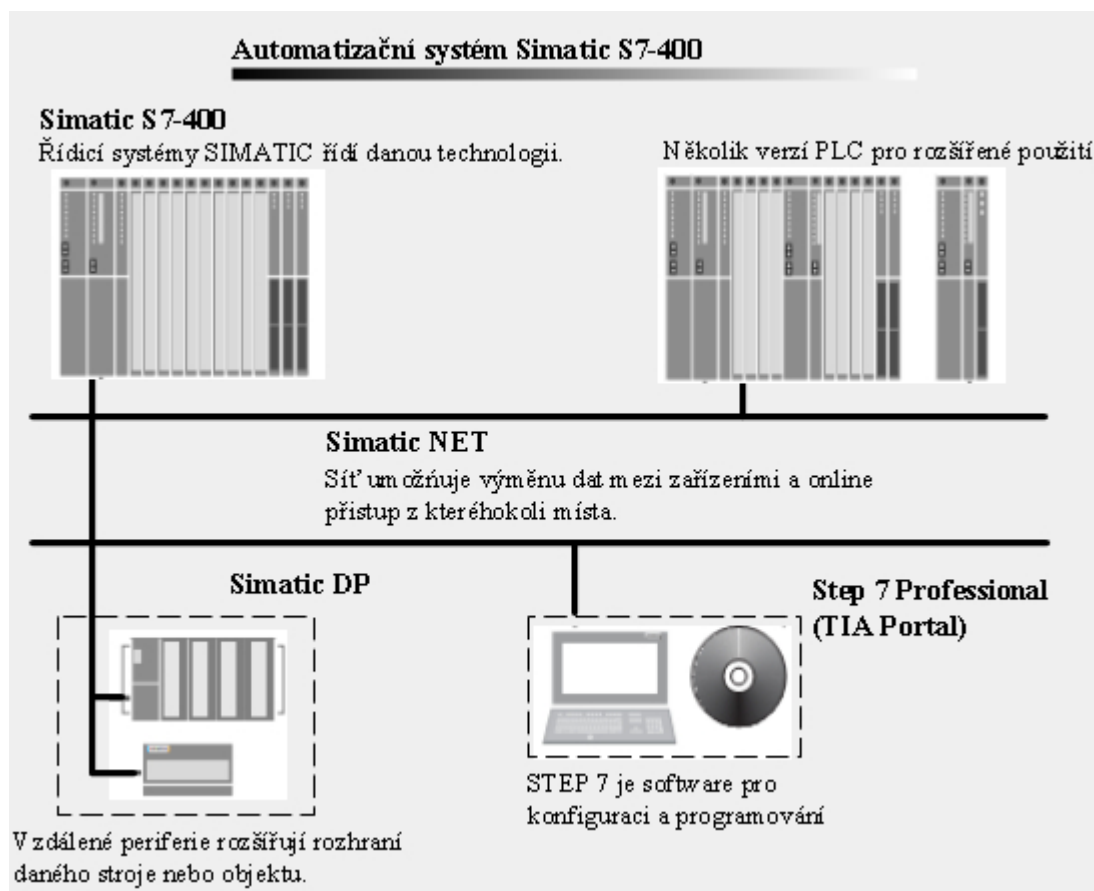
řízeného systému. Cyklické přerušení má pak smysl tam, kde události probíhají cyklicky se stanovenou časovou periodou. Jakmile je přerušení obslouženo, program pokračuje od místa vzniku přerušení. [1][4]



Obrázek 2.1: Cyklické provádění uživatelského programu u Simatic S7-400 [1]

Zasazení PLC Simatic S7-400 do širšího konceptu průmyslové automatizace je patrné z obrázku 2.2, kde kromě samotného PLC vidíme i dílčí prvky tvořící celkový automatizační systém. Jednou z nejvýznamnějších součástí tohoto automatizovaného systému pak jsou distribuované periferie Simatic DP, které umožňují rozšíření rozhraní pro řízení daného systému či technologie. V případě projektu řízení sekundárního odprášení hraje distribuovaná periferie významnou roli a její detailní implementace a konfigurace bude rozebrána později v kapitolách HW konfigurace PLC.

Vzhledem k dopadům na řízení jsou vzdálené vstupně výstupní periferie integrální součástí každého automatizačního systému, jelikož umožňují rozšíření rozhraní řízené technologie a její napojení na daný komunikační protokol. V tomto případě je použita komunikace pomocí průmyslového standartu PROFIBUS DP, nicméně lze použít i ostatní komunikační protokoly jako PROFINET IO apod. V obou případech je zavedení těchto komunikačních standardů ve formě subnetu, tedy sítě, po které může probíhat výměna dat mezi zařízeními tvořícími výsledný automatizační systém. Nutno podotknout, že celý automatizační systém je taktéž obohacen o uživatelský interface v podobě Simatic HMI (Human Machine Interface) nebo jiného vizualizačního systému sloužícího k zobrazení řídicích pochodů daného systému a interakci s obsluhou.



Obrázek 2.2: Automatizační systém s PLC Simatic S7-400 [1]

Známe-li tedy celkovou zamýšlenou koncepci výsledného automatizačního systému je možno přistoupit k popisu samotného PLC Simatic S7-400 a jeho modulů, tvořící ústřední součást celé řídicí koncepce. Na obrázku 2.3 je zobrazeno PLC Simatic S7-400 sestávající z CPU, zdroje, technologických, komunikačních modulů apod. V návaznosti na to můžeme říct, že každé PLC řady S7-400 se může skládat z:

- Lišty pro uchycení všech modulů (Rack)
- Zdroje napětí (Power supply)
- Mikroprocesoru (CPU)
- Modulu komunikačních rozhraní (Interface module)
- Signálových I/O modulů
- Modulů speciálních funkcí (Vysokorychlostní čítače, PID apod.)
- Komunikačních modulů (komunikační preprocesory CP)



Obrázek 2.3: PLC Simatic S7-400 [1]

Celkové složení vždy odpovídá řešené problematice, tzn., pokud je přítomna regulace, je možno v hardwarové konfiguraci PLC najít modul určený pro regulaci pomocí PID, jsou-li kladeny vysoké požadavky na rychlost čítání určité fyzikální veličiny v procesu, je do HW konfigurace zahrnut vysokofrekvenční čítač apod.

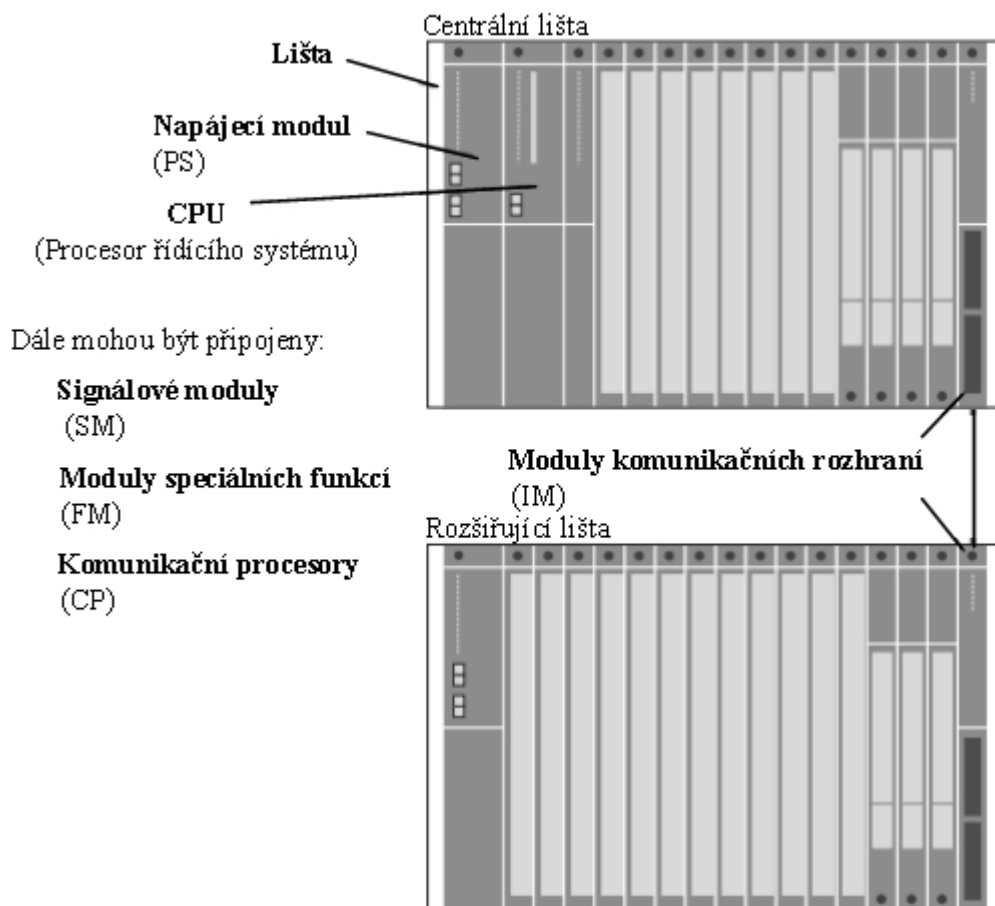
Mikroprocesor (CPU) je chápán jako řídicí inteligence automatu, která obsluhuje všechny programované instrukce a systémové služby a zajišťuje diagnostiku a komunikaci napříč automatizačním systémem. Hlavními rozdíly mezi jednotlivými typy CPU je velikost paměti a rychlost zpracování instrukcí tzn. délka cyklu. Modul jako takový taktéž poskytuje informaci o tom, v jakém je PLC provozním stavu (RUN, STOP, MAINTANANCE) a signalizační led diody ukazují, jestli není chyba v HW konfiguraci, interní chyba v programu nebo porucha komunikace (External, Internal Fault, Bus Fault). Jsou zde taktéž všechna komunikační rozhraní, který daný typ CPU podporuje – MPI, DP, MPI/DP a PN. [6]

Signálové moduly (SM) jsou vstupně/výstupní moduly pro zpracování signálů z technologie. Pro zpracování diskretní signálů jsou určeny digitální moduly a pro zpracování analogových veličin jsou určeny analogové moduly, přičemž každých z typů modulů má odlišné vlastnosti z hlediska počtu měřicích kanálů. V případě digitálních modulů jsou odlišnosti zejména dle výsledného počtu použitelných vstupů a výstupů.

Moduly speciálních funkcí (FM) jsou moduly inteligentně zpracovávající vstupní signál, bez závislosti na CPU. Pověšinou se jedná o úlohy, které CPU není schopno dostatečně rychle vykonat, jako je vysokorychlostní čítání pulzů, ovládání motorů, polohování apod.

Komunikační moduly (CM) zařizují komunikační úlohy CPU, přičemž se jedná o fyzické připojení k partnerovi a zajištění korektní datové výměny včetně případné diagnostiky. Jedná se např. o moduly nabízející komunikaci na bázi standardů Industrial Ethernet, PROFINET, PROFIBUS apod. V našem případě je použit komunikační modul CP 443-1, což je modul zajišťující komunikaci na bázi Industrial Ethernetu. [1][4]

Komponenty řídicího systému Simatic S7-400



Obrázek 2.4: Komponenty řídicího systému Simatic S7-400 [1]

V souvislosti s tím je možno uvést kompletní PLC konfiguraci, která bude použita pro řízení sekundárního odprášení a která odpovídá výše uvedenému teoretickému rozboru. Jak lze poznat z obrázku 2.5 PLC obsahuje CPU, dva komunikační moduly CP a jeden interface modul IM. Detailnější popis komponent je pak uveden v kapitole 6, v části zabývající se návrhem HW konfigurace v prostředí STEP7. Nyní pouze poznamenejme, že toto PLC, spolu se vzdálenou periferií ET200S tvoří výsledný automatizační systém.

Poslední neméně důležitou částí týkající se PLC S7-400 je struktura paměti CPU, reakční časy a dovolené doby cyklu při vykonávání uživatelského programu. To vše je uvedeno v [6], spolu s výpočtem všech těchto časů a popřípadě i s kalkulací reakčních časů při vykonávání přerušení apod. Informace o těchto hodnotách jsou pak klíčové, abychom předešli chybám plynoucím z překročení dovolených časů cyklického provádění programu.



Obrázek 2.5: Výsledné PLC S7-400 použité pro řízení technologie

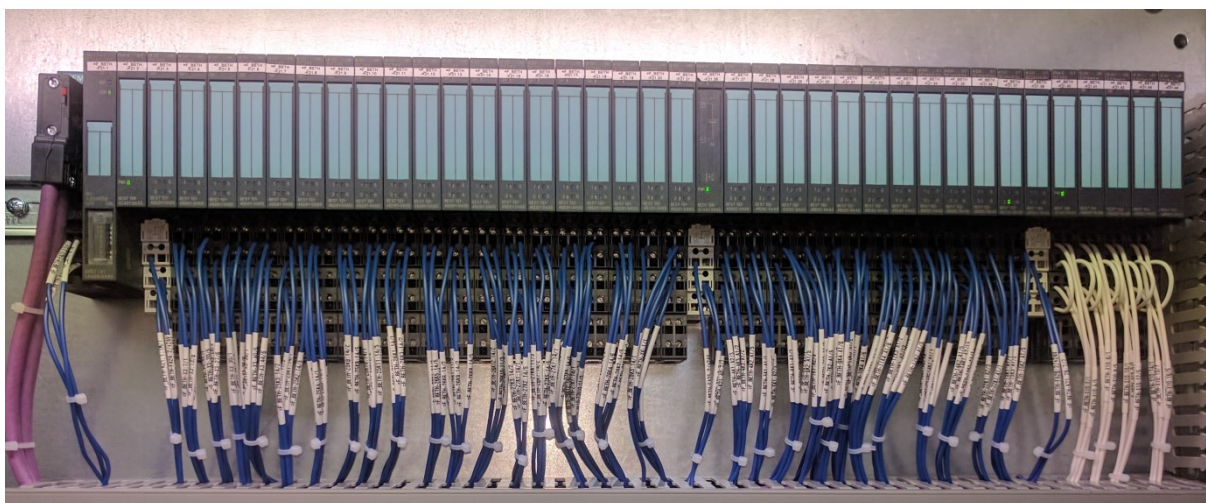
2.2 Distribuovaná periferie ET200S a PROFIBUS DP

Jak už bylo v předcházející kapitole uvedeno, distribuované periferie umožňují rozšíření rozhraní řízené technologie, což v praxi znamená, že můžeme s distribuovanými vstupy a výstupy zacházet úplně stejně jako s centrálními vstupy a výstupy, jež se nacházejí přímo na řízené technologii. Podobně jako v případě CPU rozlišujeme rozličné produktové řady, v případě distribuovaných periférií rozlišujeme jednotlivé typy dle IP krytí, rychlosti datové výměny apod. Konkrétně se jedná o typy ET 200L, ET200M, ET200S, ET200R a další typy pro speciální využití. [7]

V případě projektu řízení sekundárního odprášení je použita distribuovaná periferie ET200S jakožto univerzální periferie se stupněm krytí IP20. Základní konfigurace je možno vidět na obrázku 2.6 a úplná konfigurace, která je reálně použita pro řízení, je na obrázku 2.7. Kompletní konfigurace obsahuje všechny vstupní a výstupní karty i s propojením na příslušný akční člen technologie. Jak je možno z obrázku 2.7 taktéž poznat, je použit komunikační standard PROFIBUS DP, který je založen na standardu IEC 61158//61784. Modul komunikačních rozhraní je potom vnímán jako DP master a dílčí moduly, které jsou připojeny k subnetu, jsou vnímány jako DP slaves.



Obrázek 2.6: Distribuovaná periferie ET200S [7]



Obrázek 2.7: Kompletní distribuovaná periferie ET200S použita pro řízení technologie

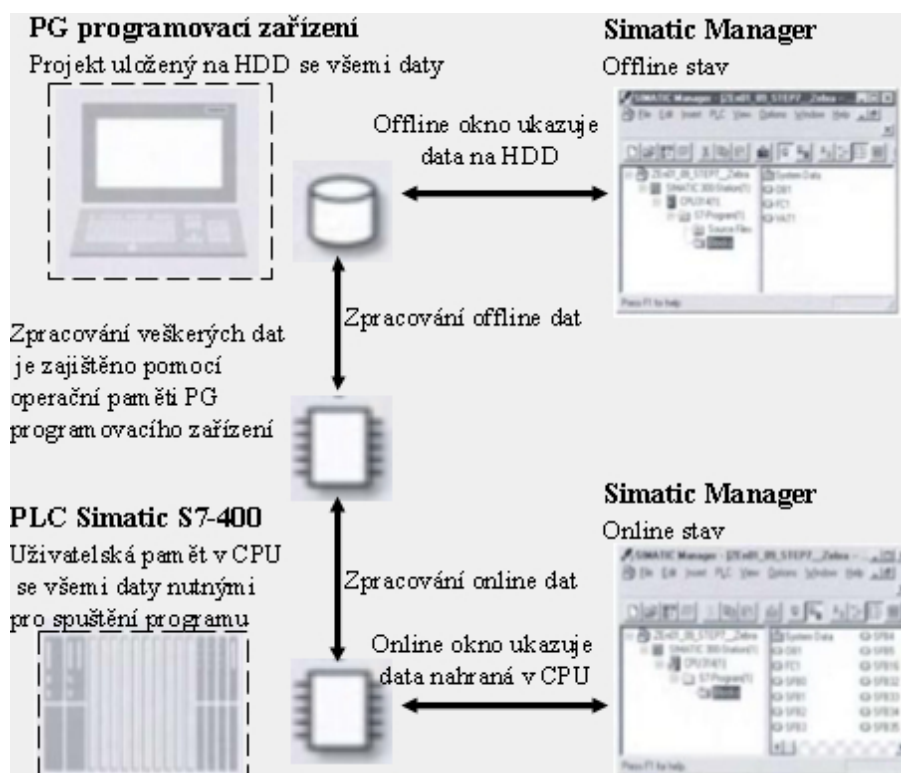
Ačkoli je idea master-slave komunikací všeobecně známa, je vhodné poukázat na dané funkcionality ve vztahu k tomuto automatizačnímu systému a určit jednotlivé úlohy a jejich vykonávání.

DP master modul je aktivní stanice a vykonává cyklickou výměnu dat s DP slave moduly, které lze označit jako pasivní stanice z pohledu datové výměny. DP slave moduly sestávají v mnoha případech z vstupně/výstupních technologických signálů. Tento subnet tvořený moduly master a slave je označován jako PROFIBUS DP master systém, přičemž aktualizování hodnot v master modulu je závislé od počtu slaves modulů. Jelikož se jedná o „mono-master“ systém je ve většině případů přítomen pouze jeden master modul. V případě tohoto automatizačního systému je přítomen jeden master komunikační modul a jeden slave, který představuje vzdálená periferie s fyzickými vstupy a výstupy technologie sekundárního odprášení.

2.3 Programovací prostředky PLC Simatic S7-400

Programování funkcí uživatelského programu v PLC můžeme zajistit pomocí několika programovacích jazyků, které mají definované parametry napříč celým spektrem programovatelných automatů. Ve zkratce se jedná o jazyky LAD, FBD, STL, SCL a GRAPH, přičemž každý z těchto jazyků poskytuje lehce odlišné možnosti při programování výsledného řídicího algoritmu. V této práci nebudou rozebírány všechny tyto jazyky, ale pouze ty, jež budou použity pro implementaci řídicích algoritmů pro technologii sekundárního odprášení. Nutno podotknout, že detailní rozbor programových celků bude uveden později, v kapitolách popisující návrh samotných algoritmů.

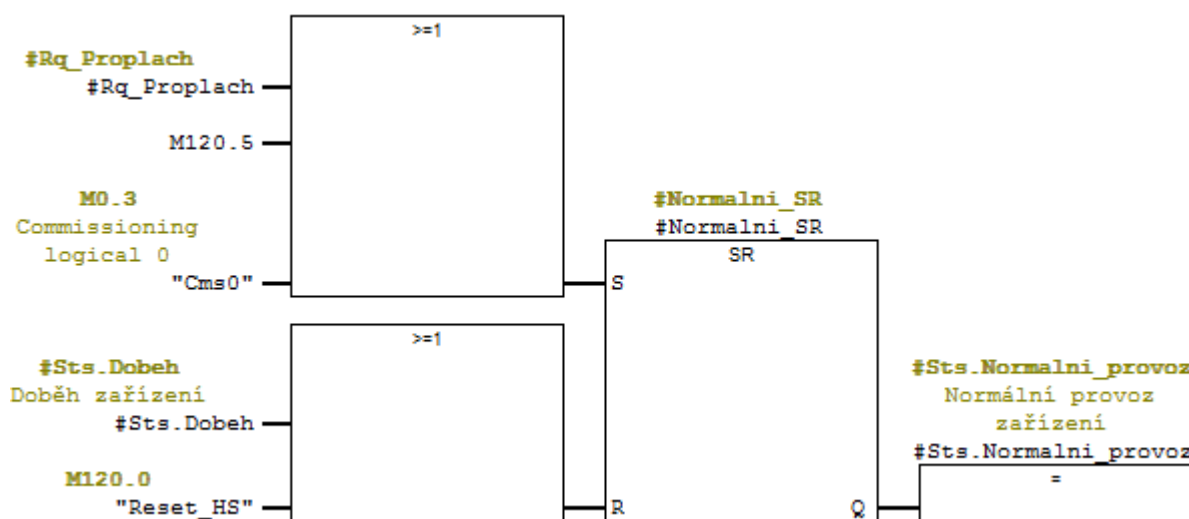
Prostředí, jež nám umožňuje plnohodnotnou práci z celkovým automatizačním systémem Simatic S7-400 a jinými se nazývá STEP7. Jedná se o software, který je předchůdcem komplexního programového řešení pro automatizaci – TIA Portalu. STEP7 tedy umožňuje kompletní řešení komunikací, HW konfigurace včetně parametrizace a adresace modulů. To vše jde ruku v ruce s implementací uživatelského programu a obecně zpracováním dat. Na obrázku 2.8 je pak ucelený pohled na výměnu a zpracování dat v rámci automatizačního systému. Základní idea je pak přecházení mezi online módem, který nám ukazuje, co se reálně nachází v paměti CPU daného PLC a offline módem, kde připravujeme uživatelský program, konfiguruje parametry modulů apod. [4]



Obrázek 2.8: Data management v programovatelném automatu Simatic [4]

Součástí STEP7 jsou již zmíněné programovací prostředky v podobě různých programovacích jazyků. V případě tohoto projektu byly z drtivé většiny použity jazyky FBD a STL. Proto je vhodné uvést krátký přehled hlavních vlastností těchto jazyků.

Function Block Diagram (FBD) je jazykem funkčních bloků, což je grafický programovací jazyk, jehož jednoduchost a přehlednost je hlavním důvodem jeho použití napříč aplikacemi. Poskytuje velmi široké možnosti implementace logických funkcí, čímž nabízí efektivní implementaci řízení pro sekundární odprášení. [2]



Obrázek 2.9: Příklad segmentu kódu v FBD

Statement list (STL) je jazykem mnemokódů, strojově orientovaným jazykem, který je maximálně přizpůsobitelný při tvoření programu. V případě návrhu algoritmů pro sekundární odprášení je tento jazyk využit především pro konverzi časových parametrů na reálné hodnoty, aby mohli být definovány setpointy časovačů. [3]

```

L      #InValAux          #InValAux
ITD
DTR
L      2.764800e+004
/R
L      1.600000e+001
*R
L      4.000000e+000
+R
T      #Sts.InVal          #Sts.InVal

A      #Sts.ParamFlt        #Sts.ParamFlt
JC      PFlt

L      #InValAux          #InValAux
L      #InOffset           #InOffset

```

Obrázek 2.10: Příklad segmentu kódu v STL

3 Popis technologie tandemové pece a význam sekundárního odprášení

Dříve než je možno pustit se do návrhu řídicích algoritmů pro sekundární odprášení je potřeba jasně vymezit chování tandemové pece a následně definovat význam sekundárního odprášení v celé technologii. Níže v podkapitolách bude popsána funkčnost tandemové pece, její chování a její mezní stavy, s kterými je potřeba při návrhu softwaru počítat.

3.1 Tandemová pec

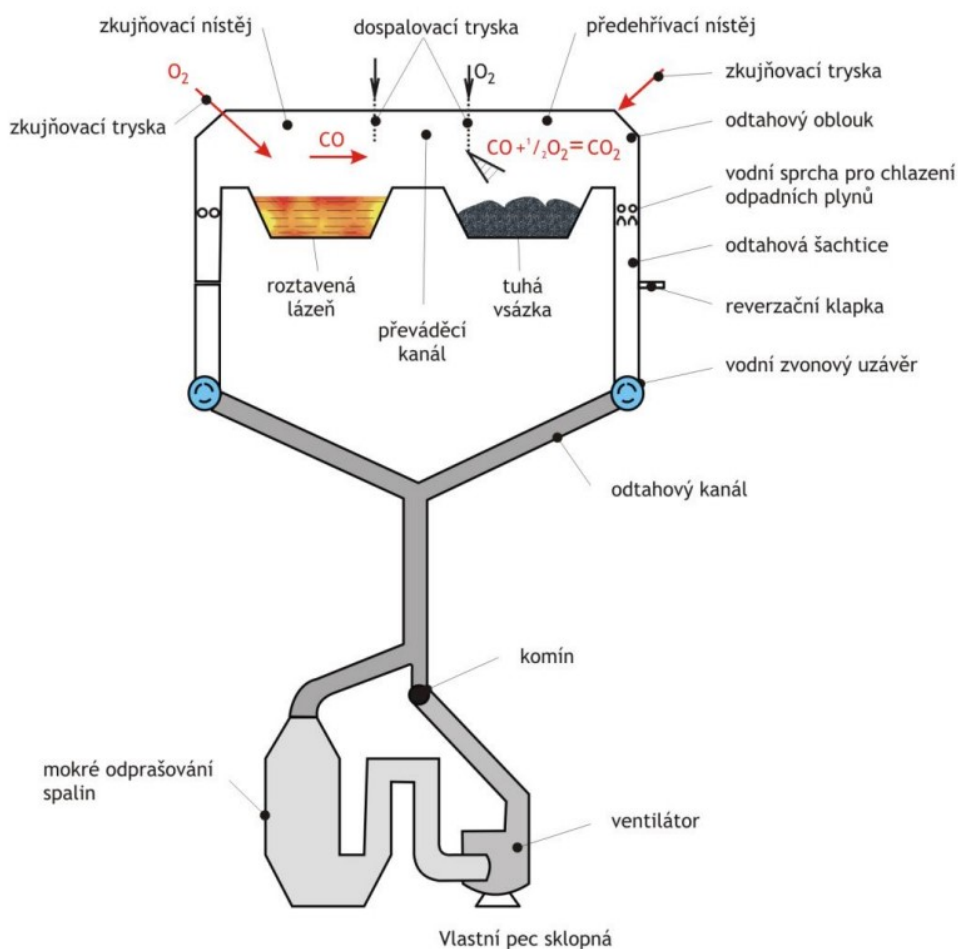
Tandemová pec se skládá se dvou nístějí, přičemž jedna slouží k přehřevu a tavení vsázky a ve druhé probíhá samotný zkujňovací proces plyným kyslíkem. Za tímto účelem jsou na stranách pece umístěny zkujňovací trysky. Jelikož v průběhu procesu dochází k vysoké produkci prachu a spalin, které jsou z části odváděny přímo z pece pomocí primárního odprášení, vzniká potřeba zajistit odprášení velkého prostoru haly ve které je tandemová pec umístěna. Každá z tandemových pecí má tedy své vlastní sekundární odprášení, jež se stará o efektivní odtah spalin z prostoru haly.



Obrázek 3.1: Tandemová pec Arcelor Mittal a.s. [9]

3.1.1 Základní charakteristika technologického celku

Samotnou pec pak tvoří dvě obdélníkové sklopné nístěje uzavřené klenbou. V přední stěně každé nístěje je sázecí okno a na obou stranách pece odtahové kanály. Pod těmito kanály je strusková komora. Po obou stranách jsou zkujňovací trysky (primární), které přivádějí do pece vzduchu. Svisle jsou pak vedeny dospalovací trysky (sekundární) pomocí kterých je přiváděn poměr vzduchu a kyslíku pro dodatečnou oxidaci oxidu CO.



Obrázek 3.2: Schematické znázornění struktury tandemové pece [5]

Jak je z obrázku pěkně vidět odtah spalin je zajištěn pomocí primárního a sekundárního odprašení, přičemž primární odprašení je realizováno odtahem přímo z pece a sekundární neboli mokré odprašení je realizováno jako odtah spalin z haly, ve které se tandemová pec nachází. Jedná se tedy o dva nezávislé technologické celky zajišťující efektivní odprašení celé technologie tandemové pece. [5]

3.1.2 Popis procesu vzhledem k řízení odprášení

Jelikož řízení sekundárního odprášení do značné míry závisí na procesech probíhajících v tandemové peci je vhodné si je ve skutečnosti vyjmenovat a uvést jejich význam ve vztahu k chodu odprášení.

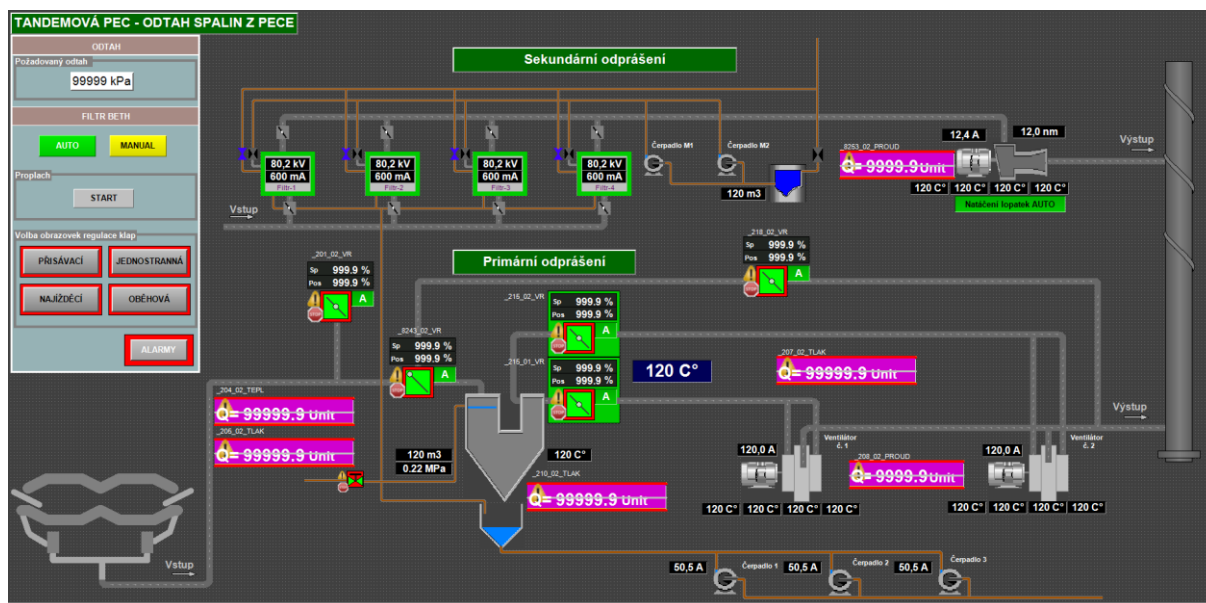
Posloupnost procesu tedy můžeme definovat:

1. Zvážení vsázky po předchozím odpichu
2. Kontejner doveze ocelový šrot
3. Prohřev šrotu (10% doby zkujňování)
4. Zavážení vápna jako příměsi
5. Pokračování prohřevu
6. Nalítí surového železa
7. Odpich tavby ze zkujňovací nístěje
8. Započetí zkujňovacího procesu v předeřhřivací nístěji
 - 8.1. V jedné třetině doby je stohována první pěnivá truska
 - 8.2. Přidání vápna a ředidel – vytvoření nové strusky
 - 8.3. Úprava vlastností oceli (pomocí přísad nebo změnou dmýchání kyslíku pomocí pojezdu trysek)
9. Odpich ve zkujňovací nístěji – celý technologický proces se obrátí

Jak bude později ve funkční analýze uvedeno, po každém odpichu je proveden oplach jednoho pole elektrického filtru, proto je zde na místě uvést tento proces odprášení do širšího kontextu chodu tandemové pece, aby bylo zřejmé, které změny v technologii vedou k inicializaci oplachu. Signál nesoucí informaci o odpichu pece je přiveden do PLC a na základě tohoto signálu je spuštěna příslušná sekvence oplachu.

3.2 Sekundární odprášení tandemové pece (BETH elektrofiltr)

Jak již bylo zmíněno výše, jedná o odprášení, které slouží k odtahu spalin z pece z prostředí haly, ve které se pec nachází. Součástí Ocelárny jsou čtyři tandemové pece – tandemová č. 2, č. 4, č. 6 a č. 8. Každá z těchto pecí má své primární a sekundární odprášení. Na obrázku 3.3 je pak možno vidět schematické zakreslení primárního a sekundárního odprášení ve vizualizaci pro tandemovou pec č. 6.



Obrázek 3.3: Schematické zakreslení primárního a sekundárního odprášení ve vizualizaci pro tandemovou pec

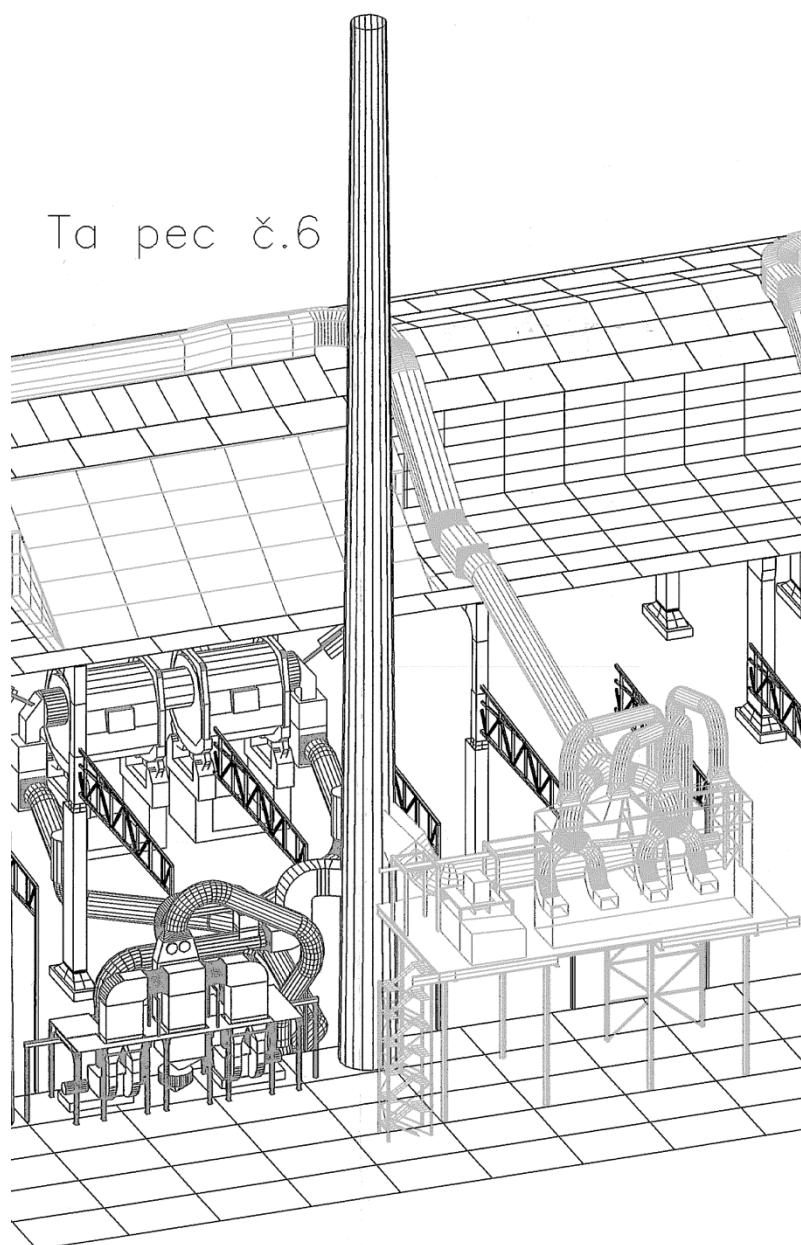
Co se týče konkrétně sekundárního odprášení, jehož řízení je předmětem celé práce, můžeme říct, že se jedná o mokrá elektrofiltr značky BETH Ricowerk. Jedná se o elektrický filtr s automatickou regulací napětí pomocí vody oplachu a mlhy. Slouží k čištění průmyslového odpadního vzduchu a spalin při teplotě, která leží v blízkosti rosného bodu plynu.

Tento filtr sestává ze 4 paralelních polí, které slouží k filtraci znečištěného vzduchu. Každé pole filtru může být prostřednictvím předřazené klapky špinavého vzduchu a klapky čistého vzduchu z procesu vyřazeno. To značí, že každé pole filtru má s ním související motorovou klapu čistého a špinavého vzduchu, pomocí nichž regulujeme přívod špinavého vzduchu dovnitř filtru a odvádíme vyfiltrovaný, čistý vzduch ven.

Prostřednictvím potrubního systému se přivádí odpadní vzduch ze střechy haly do elektrofiltru. Zde se pomocí systému mlžných trysek vstřikuje voda, jež sytí spaliny. Působením proudu dochází k odlučování nečistot a vodních kapek, jež se kondenzují na srážecích elektrodách a společně odtékají. V určitých časových intervalech je prováděn proplach polí filtrů pomocí systému oplachových trysek. V této fázi oplachu jsou podobně jako při údržbářských pracích uzavřeny klapky spalin a čistého

vzduch před a za polem filtru. Současně je odpojeno vysoké napětí. Voda oplachu je dopravována pomocí čerpadla, přičemž klapky vody oplachu pak distribuují vodu na jednotlivá pole filtru. Dále je zde vodní mlha, která je vstřikovávána trvale i při fázi oplachu.

Detailní rozbor funkce filtru je pak uveden v následující kapitole, které popisuje celou funkčnost i s respektováním odlišných technologických stavů.



Obrázek 3.4: Schematické zakreslení primárního (tmavě šedá) a sekundárního (světle šedá) odprášení TP6

4 Funkční analýza sekundárního odprášení

Jelikož je funkce sekundárního odprášení provázána s funkcí tandemové pece lze u něj definovat několik technologických stavů, ve kterých se může nacházet a které zároveň definují odlišné způsoby řízení jednotlivých prvků, se kterých se sekundární odprášení skládá.

Následující kapitola si proto klade za cíl popsat všechny technologické stavy tak aby bylo možné na základě těchto informací vytvořit řídicí algoritmy pro běh sekundárního odprášení.

4.1 Příprava najíždění sekundárního odprášení

Dříve než je možno zapnout systém sekundárního odprášení, je nutné splnit následující podmínky:

- Ruční závěrné klapky na vedení vodní mlhy a vodního oplachu jsou otevřeny
- Vodní mlha je vstřikovávána do všech čtyřech polí
- Voda oplachu je ve fázi oplachu vstřikovávána pomocí separátních čerpadel do filtru
- Voda je k dispozici se správným tlakem a ve správném množství
- Klapky špinavého a čistého vzduchu jsou zavřeny
- Vysušení izolátorů vzduchem – aby nedocházelo k přeskokům, musí se izolátory vysušit okolním vzduchem – automatický provoz ventilátoru vzduchu oplachu po dobu 30 min.
- Na zařízení nesmí být poruchy

4.2 Najíždění sekundárního odprášení

Je-li vše výše uvedeno splněno pak je možno přistoupit k najíždění technologie podle následující sekvence:

1. Uzamykatelný spínač se nastaví do polohy „Automatik“
2. Ventilátor se zapne pomocí tlačítka
3. Po 30 minutách chodu ventilátoru je řízení pomocí automatiky uvolněno
4. Zařízení se zapne pomocí tlačítka „Anlage Ein“ (zařízení zap) – Tlačítko svítí zeleně
5. Všechna pole filtru jsou postupně vždy po dobu 24 sekund opláchnuta – čerpadlo je zapnuto a klapky vody oplachu se postupně otevřou na 24 sekund
6. Čerpadlo vody oplachu se vypne
7. Po 2 minutách se automaticky zapnou 4 agregáty vn – Hlášení na rozváděčích a na vizualizačním panelu
8. Nyní se automaticky otevře osm klap špinavého a čistého vzduchu

Zařízení filtru je v provozu a chod je automatický. Nutno podotknout že náběh i doběh se provádí zásadně v automatickém režimu technologie.

4.3 Normální provoz sekundárního odprášení

Zařízení filtru je v automatickém chodu prostřednictvím řízení rozvaděčů filtru. Definujeme tento stav jako normální provoz filtru. Normální provoz můžeme charakterizovat následovně:

- Ventilátory jsou trvale v chodu
- Voda mlhy je trvale v provozu (zespoda, kde nejsou oplachové klapky je vodní mlha, seshora na místě kde jsou oplachové klapky je přiváděna voda oplachu – idea mokrého elektrofiltru), tlak vody je snímán tlakovým senzorem na každém poli filtru a výsledek je zobrazen na vizualizačním panelu
- Čtyři jednotky elektrofiltru budou trvale provozovány současně (s výjimkou fáze oplachu) a vždy společně zapínány a vypínány
- Provoz vysokého napětí (vn sekci) je indikován na schématu
- Po každém odpichu pece je opláchnuto vždy jedno pole filtru

Sekvence oplachu pro normální režim je pak následující:

1. Zapne se čerpadlo vody oplachu
2. Čerpadlo je v chodu 45 sec a pak se zavřou klapky špinavého a čistého vzduchu prvního pole – klapky se začnou uzavírat a po dosažení koncového spínače (poloha zavřeno) vyšlou informaci o poloze zavřeno, jež je pak použita pro vizualizační účely
3. Vypne se vysoké napětí dané sekce (každé pole má svou sekci vn)
4. Klapa vody oplachu prvního pole se po 10 sec automaticky otevře – klapa se začne otevírat a po dosažení koncového spínače vyšle informaci o poloze otevřeno
5. Přivádí se voda oplachu a probíhá oplach pole
6. Po dvou minutách se klapa vody oplachu příslušného pole zavře – klapa se začne uzavírat a po dosažení koncového spínače (poloha zavřeno) vyšle informaci o poloze zavřeno
7. Dvě minuty po zavření klapky vody oplachu se opět zapne vn a klapky špinavého a čistého vzduchu prvního pole se otevřou – klapky se začnou otevírat a po dosažení koncového spínače vyšlou informaci o poloze otevřeno
8. Vypne se čerpadlo a uzavře se klapa oplachové vody pro dané pole

Po čtvrtém odpichu pece je celý filtr opláchnut a po dalším odpichu pece se začíná opět oplachem prvního pole. To značí, že po čtvrtém odpichu se vypnou všechny vysokonapěťové sekce a proběhne oplach všech polí filtru a to způsobem, že uzavřou všechny klapky čistého a špinavého vzduchu a otevřou se najednou všechny oplachové klapky. Je možno říci, že proběhne výše uvedená sekvence akorát s tím rozdílem, že se nebude jednat pouze o jedno pole ale o všechny čtyři pole. To se bude cyklicky opakovat s každým pátým odpichem tandemové pece.

4.4 Doběh sekundárního odprášení

Doběh zařízení se provádí na řídicím rozvaděči BETH v následujícím pořadí:

1. Stiskne se tlačítko „Angle Aus“ (zařízení vyp)
2. Všech 8 klap špinavého a čistého vzduchu se automaticky uzavře
3. Po jedné minutě se vypnou 4 agregáty vn a ventilátory
4. Po deseti sekundách se zapne čerpadlo vody oplachu
5. Po 45 sekundách jsou všechna pole filtru postupně oplachována vždy po dobu 24 sekund – hlášení ve schématu
6. Čerpadlo vody oplachu se vypne
7. Všechny 4 klapky vody oplachu se otevrou s cílem odvětrání vodního vedení a po 15 minutách se zavřou – hlášení ve schématu

Ruční závěrné klapky systému vodní mlhy a vody oplachu mohou být uzavřeny teprve tehdy, když voda kompletně vytekla. To zamezí zamrznutí vedení v zimě. Při zimních odstávkách by měli být zapojeny ventilátory pro sušení a čidla ohřevu v potrubí v ručním provozu.

5 Popis řízených komponent technologie

Abychom mohli navrhnout řídicí algoritmy správně je potřeba znát řízené objekty a počítat s jejich specifickými vlastnostmi. To značí, že musíme vědět, zda řízené motorové klapky mají koncové snímače, kolik poloh má každá z klap a jak je celkově umožněno její řízení. Tyto informace pak vedou na správné zvolení řídicích bloků navržených přesně pro řízení zařízení takového charakteru. Dále, v následující kapitole pak budou tyto bloky zmíněny a bude uveden jejich specifický význam.

V níže uvedené tabulce je uveden kompletní seznam komponent, jež budou zahrnuty do výsledného řízení celé technologie. Každý z těchto prvků bude podrobněji rozebrán v následujících kapitolách. Jedinou výjimkou budou ruční závěrné klapky, které ačkoli jsou technologie, nemají význam pro tvorbu řídicího softwaru, jelikož se jedná o manuálně ovládané prvky. Tyto klapky proto nemají žádné vstupní a výstupní signály, jež by PLC mohlo zpracovat a vyhodnotit určitý akční zásah.

Počet	Technologický prvek	Výrobek	Typ	Krytí
2	Ventilátor	Witt & Sohn	A-RNN8/280/355/1	IP 54
4	VN agregát	Rico-Werk	Promethos X	IP 65
4	Ruční závěrné klapky vody	Ebro	Z011 – K1	IP 65
12	Motorové klapky	Ebro	E 60 WS	IP 65
8	Tlakový senzor	Fanal	-	IP 65

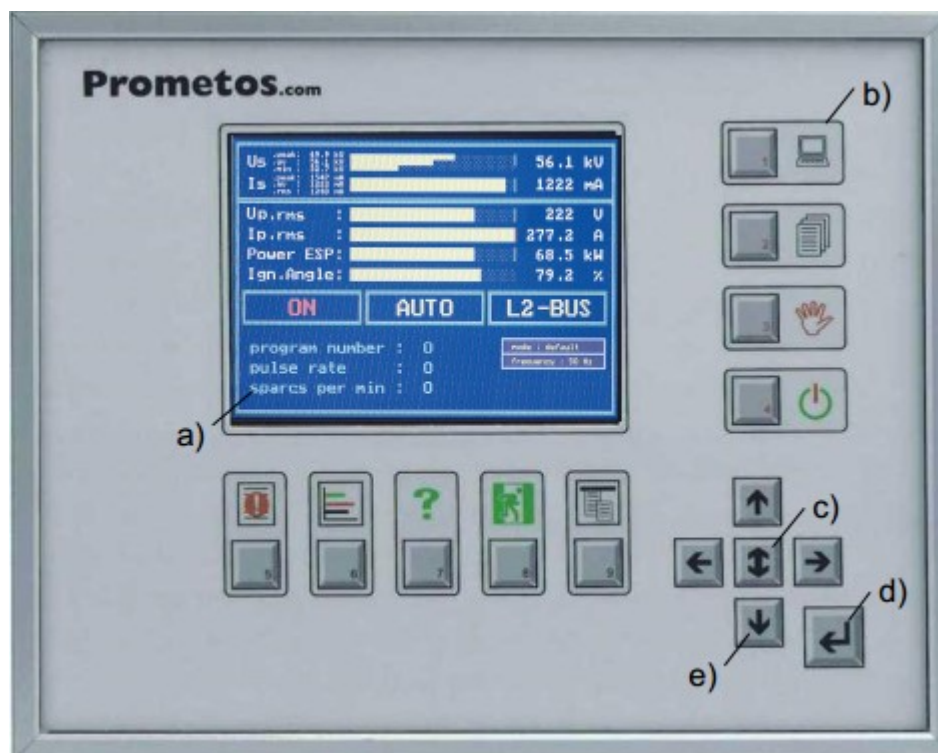
Tabulka 5.1: Seznam řízených komponent sekundárního odprášení

5.1 Vysokonapěťové agregáty Rico-Werk

Ačkoli budeme vysokonapěťové agregáty z hlediska řízení vnímat jako černé skřínky, je vhodné uvést alespoň základní význam tohoto prvku v celém systému odprášení. Tyto agregáty slouží jako zdroj vysokého napětí pro jednotlivá pole elektrofiltru. Nicméně se nejedná pouze o zdroj napětí ale taktéž o regulační systém napětí, jenž pracuje na bázi systému reálného času. Každá sekce je tedy opatřena průmyslovým real-time počítačem jenž má název Promethos.

Každý z těchto VN regulátorů lze řídit pomocí binárních signálů, které z hlediska řízení vnímáme jako digitální výstupy z PLC. Dále je zde komunikační rozhraní Profibus DP pro komunikaci s operátorským panelem nebo řídicí místností. Navíc je tento systém opatřen o možnost lokálního ovládání, pokud by selhala komunikace. Každá ze sekcí má LCD displej se všemi relevantními informacemi včetně systému pro generování poruch.

Jedná se tedy o velmi komplexní systém běžící v reálném čase, nicméně většina funkcí slouží především pro následnou obsluhu pro detailnější analýzu regulačních procesů popřípadě pro nastavení konkrétních parametrů regulace apod. Konkrétní řízení je pak realizováno pouze pomocí digitálních výstupů, jež slouží k zapínání a vypínání sekcí. Konkrétní algoritmy pro tuto část technologie budou rozebrány později v práci.



Obrázek 5.1: Průmyslový PC elektrofiltru Rico-werk

5.2 Ventilátory, Motorové klapky a tlakové senzory

Z hlediska řízení mnohem zajímavější jsou zbylé technologické prvky, jež tvoří sekundární odprášení. Těmito prvky jsou motorové klapky, ventilátory, čerpadlo a tlakové senzory. Dříve než zvolíme konkrétní způsob řízení pomocí daných funkčních bloků, je potřeba uvážit jaké možnosti pro řízení nám daný objekt nabízí. Jakmile známe, co budeme řídit, lze jednoznačně určit způsob, jakým budeme objekt řídit. Tyto funkční bloky jsou detailně popsány v kapitole zabývající se programovými prvky pro realizaci řízení. Tato podkapitola si klade za cíl uvést základní parametry všech řízených prvků, které by mohly být důležité pro výsledné řídicí algoritmy. Detailní technické specifikace jsou pak uvedeny v příslušné elektrotechnické dokumentaci celého systému.

V případě ofukových ventilátorů můžeme hovořit o klasickém diskrétně ovládaném a neregulovaném pohonu spínaném pomocí digitálních výstupů z PLC. To značí, že oba ventilátory mají řídicí signály pro zahájení chodu a zastavení. Dále je zde zpětná vazba z ventilátorů v podobě digitálního vstupu do PLC informující o chodu ventilátorů.

Motorové klapky jsou zařízení nabývající dvou možných pozic (stavů) otevřeno a zavřeno. Jelikož se z hlediska řízení jedná o zařízení se stejným způsobem řízení jako dvou-cívkový ventil, jsou vyvedeny dva řídicí signály – otevřít a zavřít. Ty jsou potom vnímány jako digitální výstupy z PLC a jsou napojeny do příslušného programového bloku konkrétní motorové klapky. Zpětná vazba od

technologie je potom realizována koncovými spínači uvnitř klapy, jež vysílají informaci o poloze klapy. Tyto signály jsou potom digitálními vstupy do PLC a umožňují programové logice vyhodnocovat, jestli je klapa otevřena / zavřena nebo se právě otevírá / zavírá. To umožní důslednou kontrolu provádění povelů, jež jsou vyslány z PLC a mají být vykonány v definovaném čase. Pokud je tento čas překročen, je generována časová porucha, značící, že povel nebyl vykonán dostatečně rychle a klapa se začala otevírat nebo zavírat se zpožděním.

Posledním prvkem, který má vliv na řízení a chod technologie jsou tlakové snímače oplachové vody a vodní mlhy. První čtyři snímače oplachové vody pak detekují, zda oplach pole filtru probíhá správně vždy v intervalech definovaných ve funkční analýze. Druhé čtyři snímače pak detekují, zdali je kontinuálně přiváděna vodní mlha pro korektní funkčnost elektrofiltru. Ačkoli mají oba typy snímačů odlišnou úlohu v technologii sekundárního odprášení, jedná se pořád o diskretní signál, který posílá informaci o stavu tlaku vody nebo mlhy do PLC. Z hlediska řízení se tedy jedná se tedy o digitální vstup.

6 Návrh řídicích algoritmů pro sekundární odprášení

Je-li stanovena funkční analýza a víme, jaké komponenty budeme řídit je možno přistoupit k jádru celé této práce. Tím je návrh konkrétních řídicích algoritmů, které budou použity pro řízení procesů uvnitř sekundárního odprášení. V této kapitole tedy bude uveden celkový postup návrhu a implementace řídicího softwaru, počínaje tvorbou hardwarové konfigurace PLC až po rozbor jednotlivých programových celků.

6.1 HW konfigurace PLC S7-400 a ET200S

Před vytvořením HW konfigurace v prostředí STEP7 je nutné udělat sumarizaci všech použitých komponent automatizačního systému použitých pro řízení. Tabulka 6.1 shrnuje komponenty PLC S7-400. Tabulky 6.2 a 6.3 ukazují všechny moduly vzdálené periferie ET200S. Co se týče PLC části, je použito CPU 414-2 PN a dále komunikační moduly – CP 443-1 jako master PROFIBUS DP a CP 441-2 jako slave DP. Poslední je pak interface rozšiřující modul IM 151-1. V tabulkách HW konfigurace ET200S jsou pak jednotlivé digitální vstupy, výstupy a analogové vstupy včetně jejich adresace.

Slot	Modul	Sériové číslo modulu	Firmware
0	PS 407 10A	6ES7 407-0KA02-0AA0	
3	CPU 414-2 PN	6ES7 414-2XK05-0AB0	V5.3
4	CP 443-1	6GK7 443-1EX30-0XE0	V3.0
5	CP 441-2	6ES7 441-2AA05-0AE0	
	IM 151-1 Standard	6ES7 151-1AA06-0AB0	

Tabulka 6.1: Hardwarová konfigurace PLC

Slot	Modul	Sériové číslo modulu	I adresa	Q adresa
1	PM-E DC24V	6ES7 138-4CA01-0AA0		
2	4 DI DC24V HF	6ES7 131-4BD01-0AB0	0.0 ... 0.3	
3	4 DI DC24V HF	6ES7 131-4BD01-0AB0	0.4 ... 0.7	
4	4 DI DC24V HF	6ES7 131-4BD01-0AB0	1.0 ... 1.3	
5	4 DI DC24V HF	6ES7 131-4BD01-0AB0	1.4 ... 1.7	
6	4 DI DC24V HF	6ES7 131-4BD01-0AB0	2.0 ... 2.3	
7	4 DI DC24V HF	6ES7 131-4BD01-0AB0	2.4 ... 2.7	
8	4 DI DC24V HF	6ES7 131-4BD01-0AB0	3.0 ... 3.3	
9	4 DI DC24V HF	6ES7 131-4BD01-0AB0	3.4 ... 3.7	
10	4 DI DC24V HF	6ES7 131-4BD01-0AB0	4.0 ... 4.3	
11	4 DI DC24V HF	6ES7 131-4BD01-0AB0	4.4 ... 4.7	
12	4 DI DC24V HF	6ES7 131-4BD01-0AB0	5.0 ... 5.3	
13	4 DI DC24V HF	6ES7 131-4BD01-0AB0	5.4 ... 5.7	

14	4 DI DC24V HF	6ES7 131-4BD01-0AB0	6.0 ... 6.3	
15	4 DI DC24V HF	6ES7 131-4BD01-0AB0	6.4 ... 6.7	
16	4 DI DC24V HF	6ES7 131-4BD01-0AB0	7.0 ... 7.3	
17	4 DI DC24V HF	6ES7 131-4BD01-0AB0	7.4 ... 7.7	
18	4 DI DC24V HF	6ES7 131-4BD01-0AB0	8.0 ... 8.3	
19	4 DI DC24V HF	6ES7 131-4BD01-0AB0	8.4 ... 8.7	
20	4 DI DC24V HF	6ES7 131-4BD01-0AB0	9.0 ... 9.3	

Tabulka 6.2: Hardwarová konfigurace ET200S – 1. část

Slot	Modul	Sériové číslo modulu	I adresa	Q adresa
21	PM-E DC24V	6ES7 138-4CA01-0AA0		
22	4DO DC24V/0,5A ST	6ES7 132-4BD02-0AA0		0.0 ... 0.3
23	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		0.4 ... 0.7
24	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		1.0 ... 1.3
25	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		1.4 ... 1.7
26	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		2.0 ... 2.3
27	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		2.4 ... 2.7
28	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		3.0 ... 3.3
29	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		3.4 ... 3.7
30	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		4.0 ... 4.3
31	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		4.4 ... 4.7
32	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		5.0 ... 5.3
33	4DO DC24V/0,5A ST	6ES7 131-4BD01-0AB0		5.4 ... 5.7
34	PM-E DC24V	6ES7 138-4CA01-0AA0		
35	2 AI 2/4WIRE HF	6ES7 134-4MB02-0AB0	500 ... 503	
36	2 AI 2/4WIRE HF	6ES7 134-4MB02-0AB0	504 ... 507	
37	2 AI 2/4WIRE HF	6ES7 134-4MB02-0AB0	508 ... 511	
38	2 AI 2/4WIRE HF	6ES7 134-4MB02-0AB0	512 ... 515	

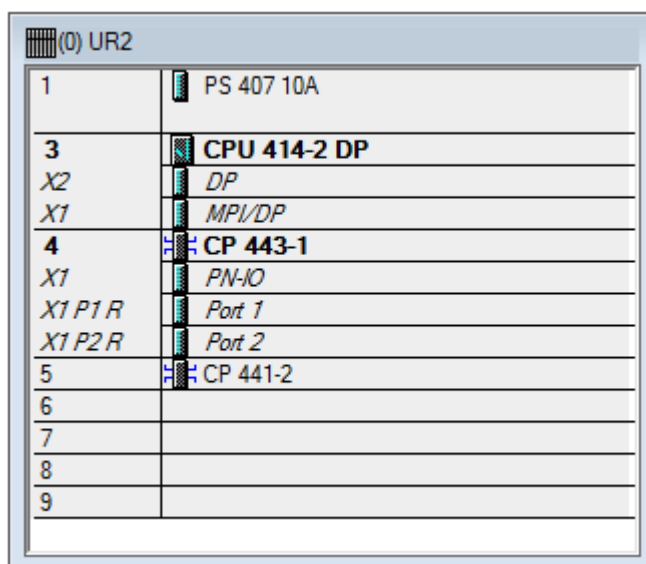
Tabulka 6.3: Hardwarová konfigurace ET200S – 2. část

6.2 Navržení HW konfigurace ve STEP7

Tvorba HW konfigurace v prostředí STEP7 je pak otázkou vyhledání příslušných komponent v katalogu a nastavení správného firmwaru vybraných modulů.

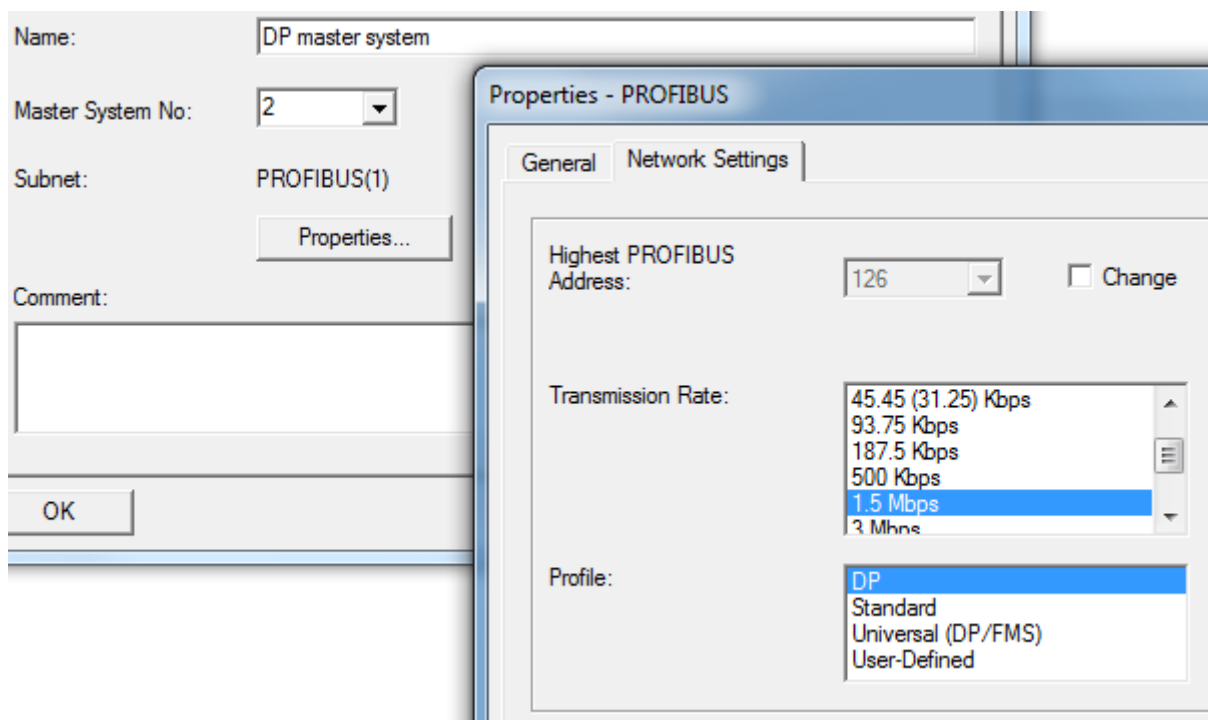
Ve výsledku to znamená, že finální HW konfigurace je tvořena napájecím zdrojem, procesorem 414-2 DP, master komunikačním modulem CP 443-1 a slave komunikačním modulem CP 441-2. U modulů, které mají rozličné verze firmware je potřeba zvolit odpovídající verzi.

Máme-li vytvořenou základní HW konfiguraci PLC je možné přidat PROFIBUS DP a nakonfigurovat vzdálenou periferii ET200S. Jelikož se bude komunikace nacházet na rozhraní X2 je možno pravým tlačítkem vybrat *Add master system* a následně jej nakonfigurovat – zadat název, pořadové číslo, přenosovou rychlost a profil subnetu. Dalším krokem je nastavení ethernetové adresy na rozhraní X1 master komunikačního modulu CP 443-1. V případě tohoto projektu byla nastavena adresa 10.32.47.60. V případě nutnosti změny rozhraní a následnou volbou mezi X2 a X1 je možno plynule přecházet pomocí funkce Add/Disconnect master system. Tento stav může nastat při potřebě testování simulace chodu technologie, kdy nechceme, aby nám chod programu ovlivňovali fyzické vstupy a výstupy na technologii. V tomto případě tedy odpojíme master systém od rozhraní X2, které slouží pro fyzické vstupy a výstupy, a připojíme jej na rozhraní X1, které je nastaveno v simulačním softwaru. Výsledkem bude, že fyzické vstupy a výstupy nebudou přepisovat stavy simulované virtuální vzdálenou periferií.



(0) UR2	
1	PS 407 10A
3	CPU 414-2 DP
X2	DP
X1	MPI/DP
4	CP 443-1
X1	PN-IO
X1 P1 R	Port 1
X1 P2 R	Port 2
5	CP 441-2
6	
7	
8	
9	

Obrázek 6.1: HW konfigurace PLC S7-400 ve STEP7



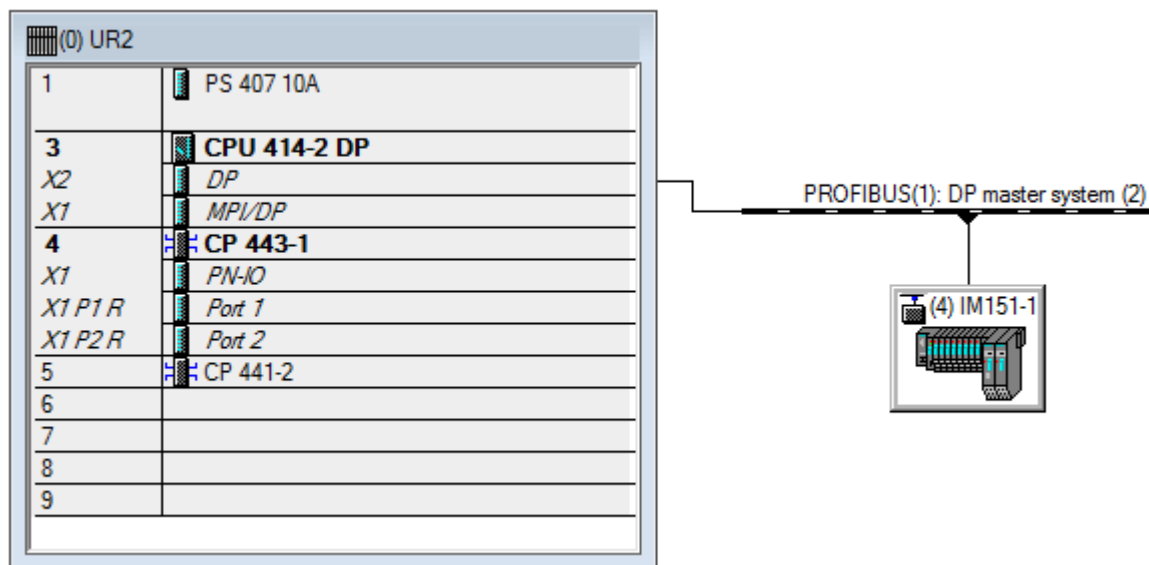
Obrázek 6.2: Nastavení parametrů PROFIBUS DP master systému

Posledním krokem je konfigurace vzdálené periferie, kdy po vložení všech digitální a analogových vstupních a výstupních modulů je nutné zajistit korektní návaznost adres jednotlivých vstupů a výstupů. Funkce *Pack Addresses* potom zajistí, aby jednotlivé adresy měli správnou návaznost, přičemž část výsledné konfigurace ET200S je uvedena na obrázku 6.3. Zde je možno sledovat správnou návaznost adres jednotlivých digitálních vstupů. To zajistí jednak správnou komunikaci se simulačním softwarem WinMOD tak následně s reálnou technologií sekundárního odprášení.

Slot	Module	Order number	I address	Q address
1	PM-E DC24V	6ES7 138-4CA01-0AA0		
2	4DI DC24V HF	6ES7 131-4BD01-0AB0	0.0...0.3	
3	4DI DC24V HF	6ES7 131-4BD01-0AB0	0.4...0.7	
4	4DI DC24V HF	6ES7 131-4BD01-0AB0	1.0...1.3	
5	4DI DC24V HF	6ES7 131-4BD01-0AB0	1.4...1.7	
6	4DI DC24V HF	6ES7 131-4BD01-0AB0	2.0...2.3	
7	4DI DC24V HF	6ES7 131-4BD01-0AB0	2.4...2.7	
8	4DI DC24V HF	6ES7 131-4BD01-0AB0	3.0...3.3	

Obrázek 6.3: Konfigurace vzdálené periferie ET200S

Výsledná podoba HW konfigurace řídicího systému užitého pro řízení odprášení je uvedena na obrázku 6.4. Je zde jak HW konfigurace samotného PLC S7-400 tak konfigurace komunikací a vzdálené periferie ET200S.



Obrázek 6.4: Výsledná HW konfigurace řídicího systému ve STEP7

6.3 Tvorba definiční listiny a I/O listu

Prvním krokem při vytváření řídicí aplikace většího charakteru je nutné vytvořit soubor symbolů, které jsou užity v rámci celého projektu. Pro tento účel slouží dva klíčové dokumenty – definiční listina a I/O list. I/O list jako soubor všech vstupů a výstupů dané technologie a definiční listina jako komplexní seznam všech symbolů a technologických prvků, které jsou součástí projektu.

Jelikož výchozí technologické řešení odprášení bylo tvořeno zastaralým řídicím systémem Simatic S5 bylo potřeba provést přehodnocení stávajících proměnných, které dle nové elektrické dokumentace již dále nebudou potřeba, a proto budou jejich vstupy a výstupy zrušeny. To vše se zohlednilo v I/O listu, který udává výslednou podobu vstupu a výstupu, jež jsou použity v rámci celého řídicího softwaru.

I/O list je uveden v příložených dokumentech k této práci pod názvem *AC3247_IO_list.xlsx*, kdy AC3247 udává značení řešeného projektu v rámci firemní struktury. Samotný soubor je pak tvořen několika částmi (kartami) – digitální vstupy, digitální výstupy, analogové vstupy, simulované digitální vstupy, simulované digitální výstupy a simulované analogové vstupy. Příklad struktury je pak uveden na obrázku 6.5. V případě technologických vstupů a výstupů jsou uvedeny jednotlivé bitové adresy, v případě simulovaných vstupů a výstupů jsou kromě fyzické bitové adresy uvedeny i korespondující adresy driveru simulace. To zajistí korektní komunikaci mezi simulací a PLC.

Symbol	Adresa	typ	Popis starý
HS1_Flt	I0.0	BOOL	HS1 porucha
HS1_Rung	I0.1	BOOL	HS1 chod
HS2_Flt	I0.2	BOOL	HS2 porucha
HS2_Rung	I0.3	BOOL	HS2 chod
HS3_Flt	I0.4	BOOL	HS3 porucha
HS3_Rung	I0.5	BOOL	HS3 chod
HS4_Flt	I0.6	BOOL	HS4 porucha
HS4_Rung	I0.7	BOOL	HS4 chod
Press_Flush_D1	I1.0	BOOL	Tlak oplachové vody pole 1 D1
Press_Flush_D2	I1.1	BOOL	Tlak oplachové vody pole 2 D2

Obrázek 6.5: Segment I/O listu pro odprášení BETH TP6

Komplexnější celek je pak definiční listina, která zahrnuje všechny symboly a technologické celky použité v řídicí aplikaci. Tyto definiční listiny jsou pak velmi důležité pro rychlé úpravy a přehlednost celého projektu jelikož standardní postup při změně určitého parametru je v podobě změny v definičním listu, která se poté projeví v rámci programu nebo vizualizace. V rámci projektu odprášení byly vytvořeny dvě definiční listiny – *AC3247 Deflist_TP6.xlsm* a *AC3247 Deflist_MAIN.xlsm*.

Každý deflist obsahuje několik klíčových parametrů – Skupina, Symbol, Typ, Struktura a DB. Skupina je pak určitá část technologie, která pod sebou sdružuje další technologické podprvky. V našem případě to je například BETH, BETH_Field_1, BETH_Field_2, BETH_Field_3, BETH_Field_4 což značí všechny čtyři pole filtrů, kdy každé pole má své vlastní, specifické, technologické prvky jako jsou napěťové a proudové sekce, motorové klapky, ventilátory apod.

Symbol je jedinečné označení daného technologického prvku. Příkladem může být například HS1_X6_VN_I (viz. Obrázek 6.6) což je proud vysokonapěťového agregátu prvního pole. Toto označení je velmi důležité, protože stejný název je použit i pro název instančního databloku ve STEP7, název objektu a symbolů (tagů) ve vizualizaci a v elektro dokumentaci k projektu. To zajišťuje vzájemnou kompatibilitu mezi řídicím softwarem, vizualizací a elektro dokumentací v rámci řešeného projektu.

Typ objektu poté udává, o jaký funkční celek se jedná. V zásadě se rozlišují tři typy objektů – Group – což je skupina objektů, která může obsahovat další podobjekty (AnMeas, DgMeas apod.), GenDrive – funkčně složitější objekt, jehož funkční blok je použit v projektu vždy jen jednou a pak ostatní jednoduché objekty, které jsou v projektu použity na více místech. Z obrázku 6.6 je opět vidět, že v případě tohoto projektu se jedná například o objekty DrvOnOff, AnMeas, DgMeas apod. Všechny tyto technologické objekty jsou velmi významné a udávají jakým způsobem je řídicí software tvořen. Proto jim bude dále v kapitolách návrhu algoritmů věnována velká pozornost. Zde pouze

poznamenejme, že se jedná o funkční bloky všech ventilů, pohonů a dalších technologických prvků, které jsou vytvořené firmou Ingeteam a.s. a tvoří základ každého řídicího softwaru.

/BETH			
BETH	BETH	Group	BETH
BETH_Common	Fan_Pur_Air_M13	DrvOnOff	DrvOnOff
BETH_Common	Fan_Pur_Air_M14	DrvOnOff	DrvOnOff
BETH_Common	Flush_Pump_M20_M21	DrvOnOff	DrvOnOff
BETH_Common	Fan_Blade	DrvOnOff	DrvOnOff
BETH_Field_1	BETH_Field_1	Group	BETH
BETH_F1	_HS1_X6_VN_I	AnMeas	AnMeas
BETH_F1	_HS1_X6_VN_U	AnMeas	AnMeas
BETH_F1	Press_Flush_D1	DgMeas	DgMeas
BETH_F1	Press_Flush_D5	DgMeas	DgMeas
BETH_F1	HS1	DgMeas	DgMeas
BETH_F1	InVlv_Cln_Air_M1	VlvTwoCoil	Vlv
BETH_F1	InVlv_Drt_Air_M5	VlvTwoCoil	Vlv
BETH_F1	InVlv_Flush_Wattr_M9	VlvTwoCoil	Vlv

Obrázek 6.6: Segment definičního listu pro odprášení TP6

Název skupiny struktur pak definuje jeho rozhraní s ostatními částmi programu a vizualizací. Tyto struktury jsou ve STEP7 ve formě UDT bloků, jež definují interface daných programových struktur. Velmi podstatné jsou čtyři struktury signálů – žádané hodnoty a parametry (SP – setpoint), povely z HMI stanice (CMD – command), aktuální stav zařízení (STS – status), aktuální poruchy a varování (FLT – fault). Tyto struktury se poté velmi často objevují v programu a poskytují nezbytné informace o technologických zařízeních, které chceme řídit. Pro ilustraci je vhodné uvést příklad statusu motorové klapky, jejíž signál může nabývat statusu otevřeno, zavřeno, otevírání, zavírání apod. Struktury lze tedy označit jako pomyslné třetí rozměry technologického signálu, který je zpracováván řídicím systémem.

Posledním parametrem je konkrétní datový blok (DB) jež má každá technologická skupina. V tomto případě se jedná o instanční DB daného objektu, ve kterém jsou uložena všechna data instance tohoto objektu – stavy, parametry, poruchy, interface HMI apod. Číslo DB, které je uvedeno v deflistu, je potom stejné i v programu STEP7 (viz. Obrázek 6.7).

BETH	1100	FB800	L3_1_Data.FB
DrvOnOff		FB810	L3_2_Data.FB
DrvOnOff		FB1100	BETH_Common.FB
DrvOnOff		FB1110	BETH_Field_1.FB

Obrázek 6.7: Definiční listina a STEP7

6.4 Programová struktura

Jak bylo v předcházející kapitole popsáno, každá definiční listina sestává se specifických technologických objektů, jež umožňují konkrétní realizaci řídicích algoritmů ve STEP7 prostředí. Nicméně pokud chceme vytvářet řídicí algoritmy tato samotná znalost nestačí a je potřeba ji obohatit o informace týkající se obecné struktury instančních databloku, interlockových struktur apod.

Následující kapitola si proto klade za cíl postoupit v logickém sledu vytváření řídicích algoritmů a jít více do hloubky sofistikovaného automatizačního standardu firmy Ingeteam.





6.4.1 Struktura instančního databloku

Pokud chceme pracovat s jednotlivými technologickými objekty, které budou popsány níže, je nutné znát obecnou strukturu jejich instančního databloku. Tento datablok obsahuje nejdůležitější proměnné, jež jsou řazeny do struktur a slouží nejen jako vnitřní tagy programu ale taky jako komunikační proměnné s HMI systémy. Tyto proměnné jsou potom definovány v UDT blocích uvnitř STEP7 programu.

Lze říci, že každý instanční datablok obsahuje struktury typu:

- Sp – struktura žádaných hodnot (setpointů) a parametrů z HMI systému
- Cmd – struktura povelů (commandů) z HMI
- Sts – struktura statusů daného technologického objektu
- Flt – struktura poruch (faultů) a varování
- InLck / InLck32 – struktura blokovacích podmínek (interlocků)

Nejdůležitější oblast instančního databloku je potom statická oblast, která má opět své specifické dělení na několik oblastí. První je oblast veřejná (Public) jež obsahuje proměnné, které se čtou v jiných částech programu popřípadě v jiných funkčních blocích. Z tohoto důvodu je nutné, aby tyto proměnné neměnili svůj význam a adresu. Pokud se totiž změní, programátor musí ošetřit všechna místa v programu, kde se tyto proměnné používají.

	Name	Data Type	Address
	Sts	VlvSts.UDT	100.0
	Flt	VlvFlt.UDT	108.0
	ReservedForPublicStruct	Array [112..149] Of Byte	112.0
	ReservedForPublic	Array [150..199] Of Byte	150.0

Obrázek 6.8: Statická oblast instančního DB - public

Další oblastí je HMI interface, což je rozhraní pro operátorskou stanici. Tyto proměnné, které jsou součástí této struktury, se čtou a zapisují v nadřazeném HMI systému. Opět, pokud je proveden změna

v této struktuře je potřeba danou změnu provést i v příslušné HMI stanici. Při vzniku potřeby obohatit strukturu o novou proměnnou je vyhrazen adresový prostor pod názvem ReservedForHmi.

HmiSp	VlvSp.UDT	200.0
HmiCmd	VlvCmd.UDT	224.0
HmiSts	VlvSts.UDT	226.0
HmiFlt	VlvFlt.UDT	234.0
HmiFltAck	VlvFlt.UDT	238.0
HmiOutInLckOpnSts	InLck.UDT	242.0
HmiOutInLckOpnWrn	InLck.UDT	244.0
HmiOutInLckClsSts	InLck.UDT	246.0
HmiOutInLckClsWrn	InLck.UDT	248.0
ReservedForHmi	Array [250..299] Of Byte	250.0

Obrázek 6.9: Statická oblast instančního DB – HMI interface

Poslední oblast instančního databloku je oblast soukromých (private) proměnných. Tyto proměnné se čtou a zapisují pouze v rámci daného funkčního bloku a nikde jinde v rámci programu se nepoužívají. U těchto proměnných nezáleží na změně významu či adresy jelikož nemají přesah do zbylých částí programu. Pokud tedy provedeme změnu, tato změna se projeví ve všech instancích změněného funkčního bloku.

FltNAck	VlvFlt.UDT	300.0
FltInPEM	VlvFlt.UDT	304.0
InLckOpnFB	InLckToHMI.FB	308.0
InLckClsFB	InLckToHMI.FB	342.0
DevModeFB	DevMode.FB	376.0
Change_Tof	TOF	452.0
LsOpndRchd_Tof	TOF	474.0
LsClsdRchd_Tof	TOF	496.0

Obrázek 6.10: Statická oblast instančního DB – Private

Jako poslední je vhodné uvést oblasti v daných strukturách, které jsou vyhrazené pro přidání nových proměnných bez toho, aby se změnil paměťový prostor ostatních proměnných. Tyto oblasti jsou označovány jako rezervy. V celé statické oblasti instančního databloku jsou tedy rezervované oblasti pro veřejnou strukturu, HMI interface strukturu a strukturu soukromých proměnných. Výhoda těchto rezerv tkví v tom, že je možné měnit, přidávat nebo mazat proměnné „Online“ za běhu PLC bez nebezpečí odstavení PLC a tím i řízené technologie.

Name	Data Type	Address
ReservedForIO	Array [10..99] Of Byte	10.0
Sts	VlvSts.UDT	100.0
Flt	VlvFlt.UDT	108.0
ReservedForPublicStruct	Array [112..149] Of Byte	112.0
ReservedForPublic	Array [150..199] Of Byte	150.0
HmiSp	VlvSp.UDT	200.0
HmiCmd	VlvCmd.UDT	224.0
HmiSts	VlvSts.UDT	226.0
HmiFlt	VlvFlt.UDT	234.0
HmiFltAck	VlvFlt.UDT	238.0
HmiOutInLckOpnSts	InLck.UDT	242.0
HmiOutInLckOpnWrn	InLck.UDT	244.0
HmiOutInLckClsSts	InLck.UDT	246.0
HmiOutInLckClsWrn	InLck.UDT	248.0
ReservedForHmi	Array [250..299] Of Byte	250.0
FltNack	VlvFlt.UDT	300.0
FltInPEM	VlvFlt.UDT	304.0
InLckOpnFB	InLckToHMI.FB	308.0
...		
PermClsPEM	Bool	651.1
Spare	Array [652..699] Of Byte	652.0

Obrázek 6.11: Statická oblast instančního DB – rezervované oblasti

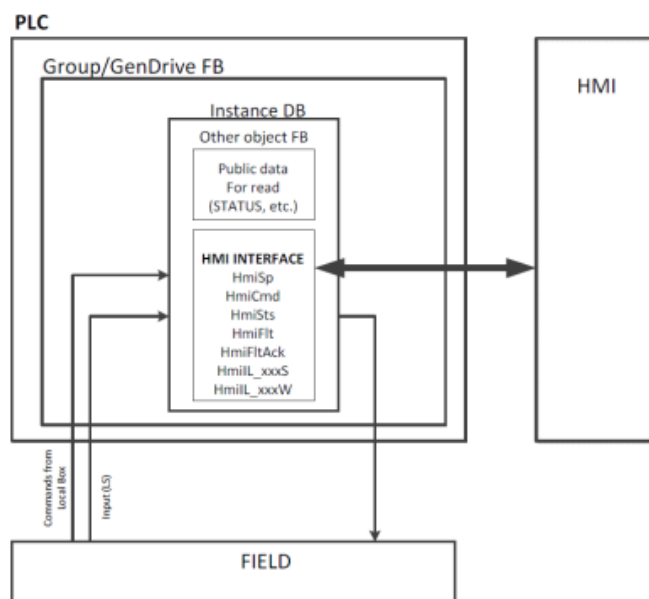
Vše výše uvedené má své opodstatnění plynoucí především z tvorby definiční listiny, která jak už bylo uvedeno, slouží jako soubor všech symbolů, struktur a technologických objektů v rámci celého řešeného projektu. To lze pozorovat i z níže uvedeného obrázku, který ukazuje kompatibilitu struktur objektu DevTwoPos s definiční listinou.

Name	Sp	Cmd	Sts	Flt
AnMeas	Byte 74 Row 14 Spare 210 UdtNo 210	Byte 2 Row 22 Spare 211 UdtNo 211	Byte 26 Row 22 InLck 212 UdtNo 212	Byte 4 Row 24 Spare 213 UdtNo 213
BETH	36 401 2000	2 747 2001	20 2877 2002	4 1009 2003
BETH_Field	36 413 2100	2 786 2101	20 2916 2102	4 1030 2103
BETH_OBRZIO	10 425 2300	2 785 2301	20 2955 2302	4 1030 2303
ConsMtr	16 150 220	2 308 221	22 864 222	4 548 223
CpuDiag	@1,@2, 16 150 220	2 175 101	2 230 102	4 150 103
DevTwoPos	@1,@2, 24 125 420	2 289 421	8 797 422	4 460 423

HmiInSp	DevTwoPosSp.UDT	200.0
HmiInCmd	DevTwoPosCmd.UDT	224.0
HmiOutSts	DevTwoPosSts.UDT	226.0
HmiOutFlt	DevTwoPosFlt.UDT	234.0
HmiInFltAck	DevTwoPosFlt.UDT	238.0
HmiOutInLckPos1Sts	InLck.UDT	242.0
HmiOutInLckPos1Flt	InLck.UDT	244.0
HmiOutInLckPos2Sts	InLck.UDT	246.0
HmiOutInLckPos2Flt	InLck.UDT	248.0
ReservedForHmi	Array [250..299] Of Byte	250.0

Obrázek 6.12: Kompatibilita statické oblasti instančního DB s definiční listinou

Pro lepší představu a sumarizaci výše uvedeného je možné uvést komplexní pohled na strukturu řízení uvnitř PLC. Níže uvedený obrázek ukazuje vzájemné propojení mezi fyzickou vrstvou (field), PLC a HMI systémem. Jak lze pozorovat, řízený technologický objekt (Group) obsahuje instanční datablok, jež obsahuje veškeré výše zmiňované struktury. Následně po zpracování programu v PLC jsou je za pomoci struktury HMI Interface zobrazovány klíčové informace na nadřazeném HMI systému.



Obrázek 6.13: Komplexní pohled na strukturu řízení uvnitř PLC

6.4.2 Implementace interlockových struktur

Nedílnou součástí každého řídicího softwaru jsou taktéž interlocky neboli blokovací podmínky, jež slouží jako bezpečnostní funkce a starají se o správný chod programových sekvencí. Obecně řečeno se jedná o podmínky, které musí být splněny, aby řízený objekt vykonal daný úkon. Typickým příkladem může být ventil, který se začne otevírat až poté, co jsou splněny všechny podmínky pro jeho otevření. To zajistí, že se ventil nebude otevírat v nevhodnou dobu, kdy by mohl například způsobit škodu na technologii.

Interlocky lze pak obecně realizovat pomocí logických funkcí OR a AND jejichž výsledek logické operace je reprezentována bitem GenInLck. Jelikož má každá instance odlišné interlocky, bývá zvykem předřazovat programový kód interlocků před samotné řídicí algoritmy pro jednotlivé technologické objekty. Do konkrétního funkčního bloku se pak přenáší výsledný bit GenInLck a datová struktura jež obsahuje aktuální stav všech definovaných blokovacích podmínek.

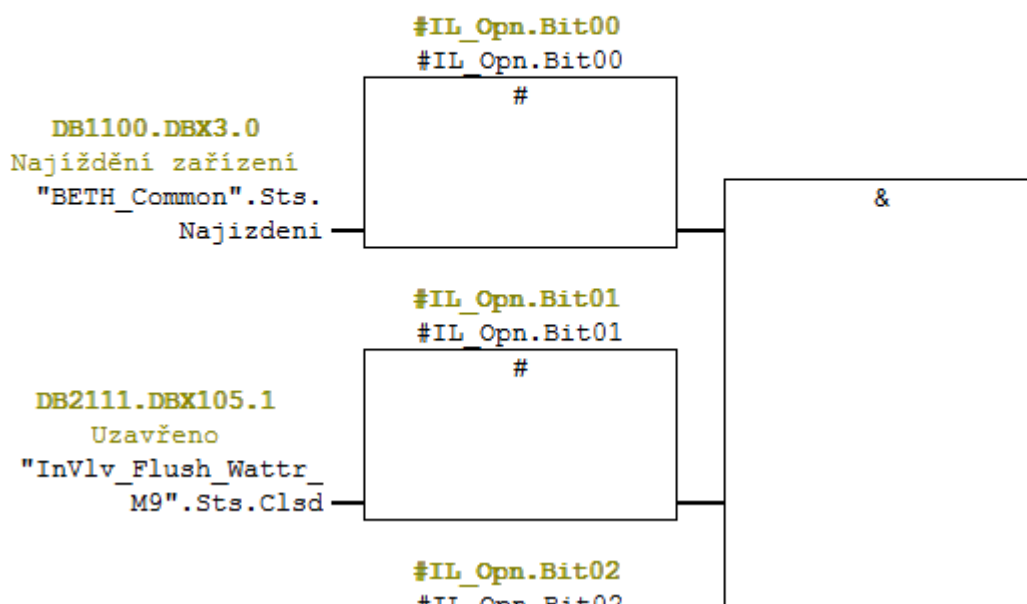
Podmínky		
&	OR	Pohon zvonu - reverzace(_023_17_POH) - V poloze Otevřeno
		Pohon zvonu - reverzace(_123_18_POH) - V poloze Zavřeno
	OR	Pohon klapy - reverzace(_023_19_POH) - V poloze Otevřeno
		Pohon zvonu - reverzace(_123_18_POH) - V poloze Zavřeno
	OR	Pohon zvonu - reverzace(_123_18_POH) - NOT Nouzový koncový snímač polohy Zavřeno aktivní
		Pohon zvonu - reverzace(_123_18_POH) - Výstup pro směr Zavřeno je aktivní
		Pohon zvonu - reverzace(_123_18_POH) - V poloze Zavřeno

Obrázek 6.14: Příklad realizace interlocků v HMI systému

Z logiky každého technologického objektu pak vyplývá jaké typy blokovacích podmínek je potřeba definovat, abychom ošetřili všechny možné stavy, jaké mohou v průběhu řízení vznikat. Obecně existují tři typy interlocků – pro start, stop a zastavení daného objektu. Pokud má tedy objekt definovány všechny tři typy interlocků, daný povel se neprovede, dokud nejsou tyto podmínky splněny. Tyto podmínky jsou povětšinou tvořeny za pomoci statusových bitů jednotlivých objektů.

▢ Network 17 : InVlv_Cln_Air_M1 - Podmínky otevření

@InLck:InVlv_Cln_Air_M1_InLckOpn



Obrázek 6.15: Příklad realizace interlocků ve STEP7

6.4.3 Kontrola vstupů a výstupů (IO Warning)

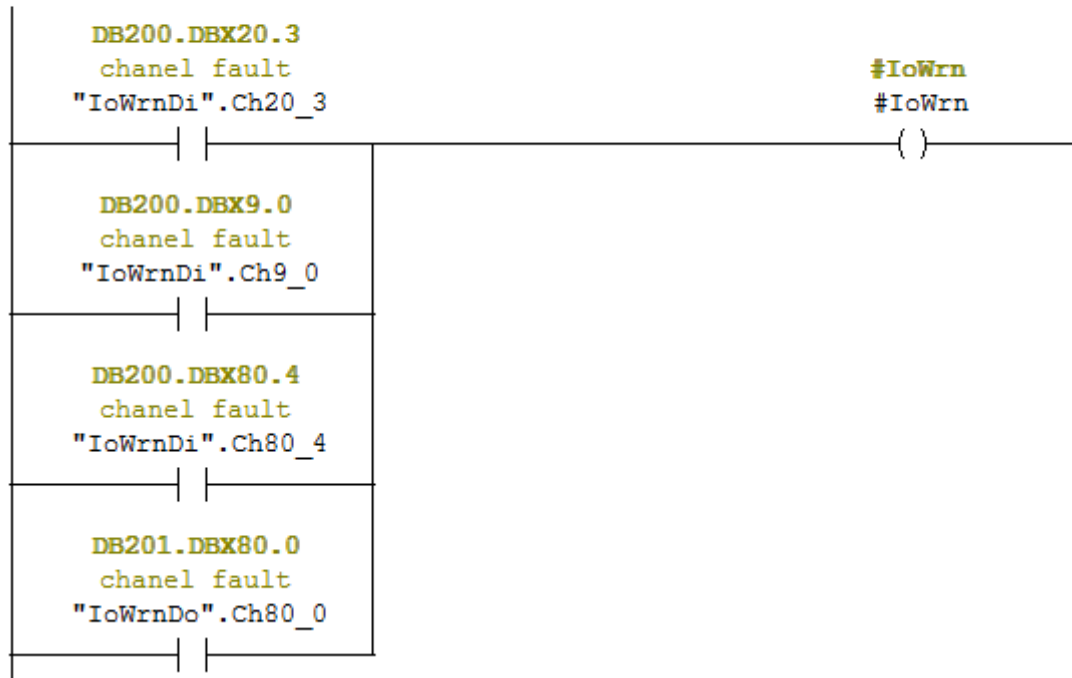
Každý z technologických objektů má vstupní parametr IOWrn, který indikuje případnou poruchu na hardwarovém vstupu nebo výstupu vstupně výstupní karty. Jedná se tedy o hardwarovou diagnostiku pro IO karty, kde jsou připojeny signály z řízené technologie.

Praktickým příkladem může být porucha na vstupu hardwarové karty I10.5. Pokud bude tato porucha odhalena, stane se status IoWrn aktivním, přičemž tento status je napojen na příslušné vizualizační prvky v HMI systému. Tímto způsobem je zajištěno, že obsluha a údržba dané technologie má detailní informace o stavu HW komponent řídicího systému.

Co se týče praktické realizace v řídicím softwaru, jsou IO Warningy definovány vždy před příslušným technologickým objektem. Digitální vstup s příslušnou adresou má svůj vlastní poruchový bit na určité adrese. Pokud tedy vezmeme v úvahu IO Warningy uvedené na obrázku 6.16, konkrétně IO Warning pro I20.3 tak tento digitální vstup má poruchový bit definovaný na adrese DB200.DBX20.3. Stejná logika pak platí i pro ostatní diagnostikované vstupy a výstupy.

□ Network 10 : Automatically generated Network with IoWrn for _016_01_V

@IoWrn



Obrázek 6.16: Příklad realizace IO Warningů

6.5 Programové prvky pro realizaci řízení

Programovými prvky rozumíme konkrétní funkční bloky, které definují vlastnosti daného objektu uvnitř sebe sama. Funkcionalita takového bloku je tedy určena programovým kódem uvnitř něj. V praxi to značí, že pokud je realizováno řízení motorové klapky pomocí funkčního bloku DevTwoPos obsahuje tento funkční blok veškeré funkce, jež jsou nutné pro komplexní řízení tohoto objektu. Těmito funkcemi jsou myšleny základní technologické funkce (otevírání, zavírání, zastavování apod.), vlastní bezpečnost, detekce poruch a sledování doby reakce na povely, funkce bezpečného vypnutí, aktuální stav (status) poskytující informaci o stavu zařízení v reálném čase apod. Výsledný funkční blok tedy udává již finální interface řízené technologie se vstupy a výstupy se kterými je možno v programu dále pracovat.

6.5.1 Obecné vlastnosti technologických objektů

Dříve než se pustíme do analýzy konkrétních funkčních bloků, dělených dle jejich specifické technologické funkce, je vhodné uvést obecné vlastnosti, jež jsou jejich společným charakteristickým znakem.

Prvním a základním prvkem každého technologického objektu jsou jeho základní technologické funkce. Tyto technologické funkce se potom liší v závislosti na řízeném objektu a tedy v závislosti na použitém funkčním bloku. V našem případě se jedná o otevírání a zavírání klap popřípadě vypínání a zapínání oplachového čerpadla. Tyto funkce tedy zajišťují základní ovládání řízeného prvku.

Druhou klíčovou vlastností všech technologických objektů je vlastní bezpečnost což je ochrana zařízení proti nesprávné akci při obdržení zmatečných povelů. To značí, že se technologický prvek, např. ventil nezačne otevírat a zavírat v tutéž chvíli, pokud obdrží oba tyto povely najednou (například nekorektním zásahem obsluhy apod.). Touto funkcí je tedy možno vyřešit např. ochranu stykačů, které je možné znovu zapnout až za definovaný čas, nebo ochranu zařízení samotného pokud má časové zpoždění při změně směru chodu a tedy velkou setrvačnost, která by jej mohla ohrozit při okamžité reakci na povel změny chodu.

Třetí vlastností je kontrola poruch a kontrola reakce na řídicí povely. V tomto případě je detekována celá řada kritických poruch vedoucích k narušení činnosti technologie. Jedná se o výpadky motorových ochranných jističů, ztráta zpětné vazby od stykačů nebo přerušení smyčky analogového měření. Při každém vzniku takového události je generována kritická chyba, jež je následně zobrazena v HMI systému. Časová porucha (Timeout) je potom specifickým druhem generované poruchy, kdy je zkoumán čas, který uplyne od vyslání povelu do jeho vykonání. Tento čas lze nastavit přímo v programu, přičemž lze obecně nastavovat čas pro dosažení koncové pozice (TimeoutReach) a čas pro opuštění výchozí pozice (TimeoutLeave). Oba tyto časy slouží jako kontrola, jestli byl povel vyslaný z PLC vykonán ve správném čase.

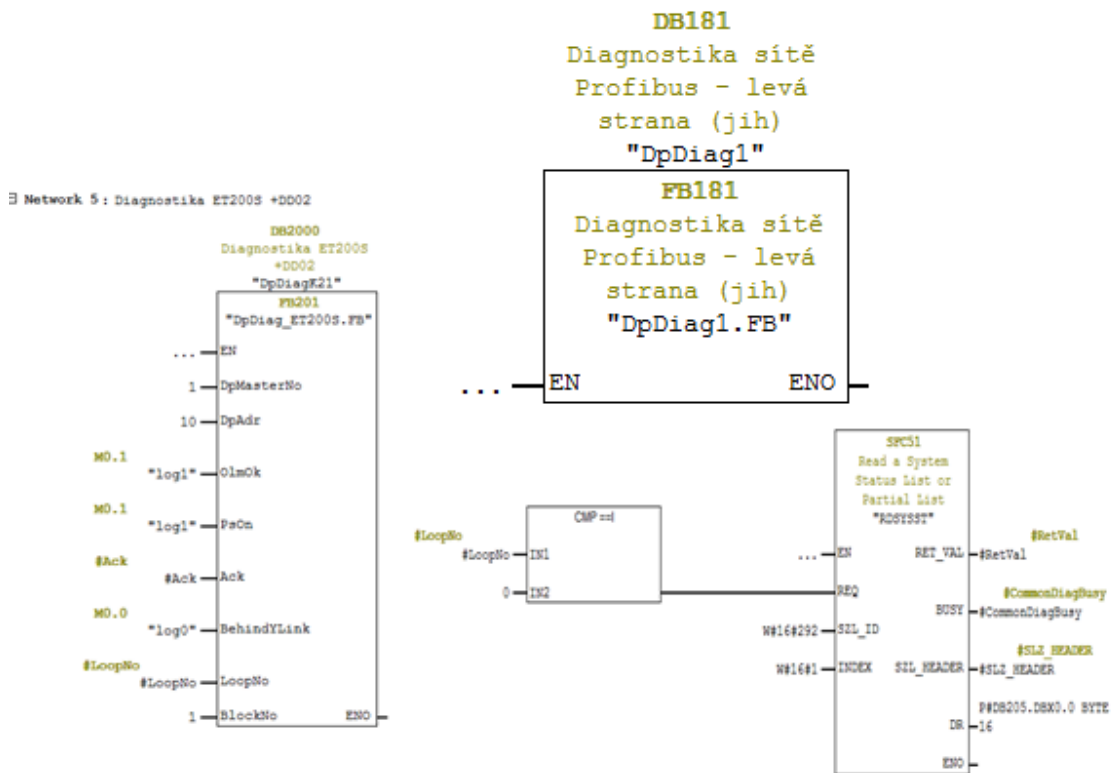
Nicméně aby byla zajištěna korektní funkce časových poruch, bylo potřeba nastavit jednotlivé časové limity pro každou motorovou klapu, která byla použita v řídicím softwaru sekundárního odprášení (viz. Obrázek 6.17). Nastavení tohoto parametru bylo provedeno pomocí vytvoření VAT tabulky přímo ve STEP7 programu. Ve VAT tabulce byl definován konkrétní adresový prostor databloku jež odpovídal proměnné TmLeave a TmReach. Jelikož defaultní hodnota těchto proměnných uvnitř databloku je 0 bylo potřeba nastavit tyto časové limity na určitou reálnou hodnotu. Pokud tato hodnota není nastavena, je automaticky generovaná časová porucha, jelikož jakýkoli čas je delší než 0s. Význam této poruchy je tedy především v kontrole řízeného objektu a slouží programátorovi jako informace o korektním vykonání povelu.

DB2109.DBD	200	"InVlv_Cln_Air_M1".HmiInSp.TmLeave	Maximální čas pro opuštění koncové polohy (s)	6.0
DB2109.DBD	204	"InVlv_Cln_Air_M1".HmiInSp.TmReach	Maximální čas pro dosažení koncové polohy (s)	7.0
DB2117.DBD	200	"InVlv_Cln_Air_M2".HmiInSp.TmLeave	Maximální čas pro opuštění koncové polohy (s)	6.0
DB2117.DBD	204	"InVlv_Cln_Air_M2".HmiInSp.TmReach	Maximální čas pro dosažení koncové polohy (s)	7.0
DB2125.DBD	200	"InVlv_Cln_Air_M3".HmiInSp.TmLeave	Maximální čas pro opuštění koncové polohy (s)	6.0
DB2125.DBD	204	"InVlv_Cln_Air_M3".HmiInSp.TmReach	Maximální čas pro dosažení koncové polohy (s)	7.0
DB2133.DBD	200	"InVlv_Cln_Air_M4".HmiInSp.TmLeave	Maximální čas pro opuštění koncové polohy (s)	6.0
DB2133.DBD	204	"InVlv_Cln_Air_M4".HmiInSp.TmReach	Maximální čas pro dosažení koncové polohy (s)	7.0

Obrázek 6.17: Nastavení maximálních časů pro koncové polohy motorových klap

S tímto bodem taktéž souvisí diagnostika sítě, konkrétně v našem případě diagnostika sítě profibus. Za tímto účelem je použit funkční blok DpDiag, který obsahuje programovou logiku pro globální diagnostiku funkčnosti komunikace, diagnostiku jednotlivých vzdálených periférií ET200S a s tím související struktury pro generování varování a poruch. Tento funkční blok je potom umístěn v hlavním organizačním bloku OB1 a je volán s diagnostikou CPU apod.

☐ Network 3: Diagnostika sítě profibus - levá strana (jih)



Obrázek 6.18: Bloky pro diagnostiku sítě Profibus

V pořadí čtvrtou vlastností objektů je paměť posledního stavu technologického objektu. Paměť posledního stavu znamená, že daný objekt si uchová aktuální stav, dokud nedostane nový povel z PLC. V praxi to znamená, že například ventil setrvává v otevřené poloze, i když už povel z HMI nebo nadřazené programové sekvence není aktivní. Tato informace je velmi důležitá s ohledem na vytváření řídicích algoritmů, jelikož je nutné posoudit vhodnost programovatelných akcí typu set, reset apod.

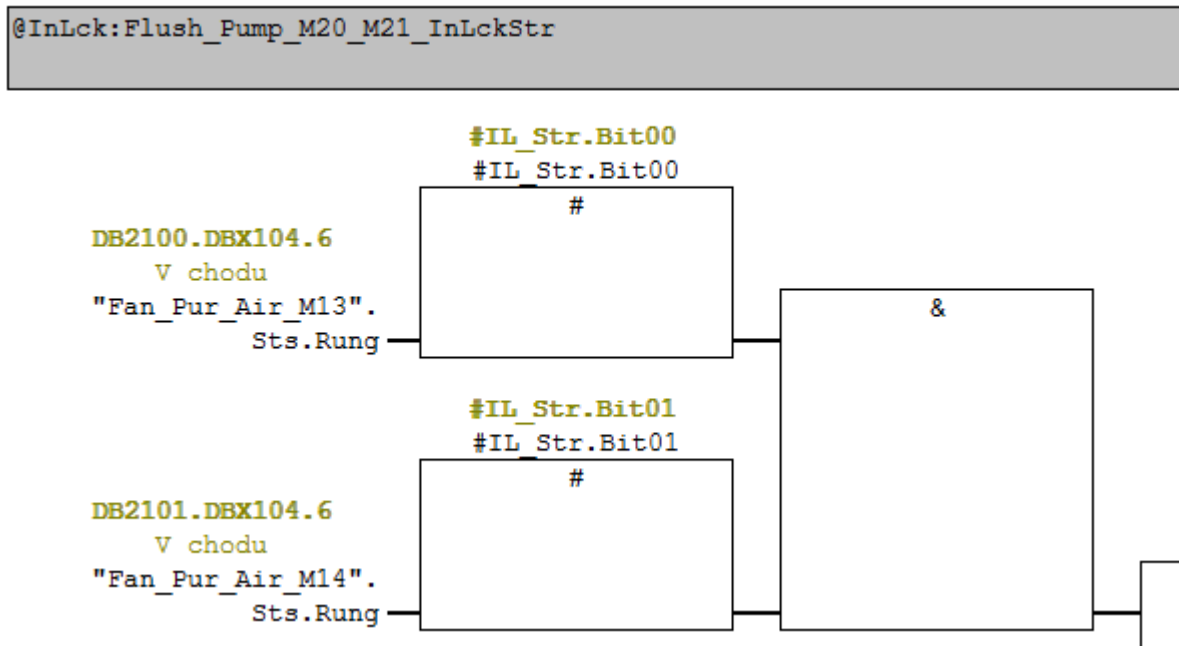
Pátou vlastností je funkce bezpečného vypnutí, jež je obsažena ve všech blocích. Každý z těchto bloků má definován „bezpečný stav“ do kterého objekt přejde v okamžiku vzniku poruchy nebo v případě aktivace nouzového vypnutí. Pokud v případě poruchy zařízení dostává objekt zmatečné signály, má vyšší prioritu ten, jež dostane zařízení do bezpečného stavu. Jednotlivé bezpečné stavy se liší objekt od objektu, ale obecně lze říci, že např. jedno cívkový normálně otevřený ventil má bezpečný stav otevřeno, jedno cívkový normálně uzavřený ventil uzavřeno a například dopravníky nebo čerpadla mají bezpečný stav zastaveno nebo vypnuto. To vše zajistí minimalizaci případných škod na technologii při nestandardním chování řízených objektů.

Poslední, šestou klíčovou vlastností všech technologických objektů jsou informace o aktuálním stavu neboli statusu. Každý objekt obsahuje ve svém instančním databloku strukturu statusů (Sts) ve které jsou definovány proměnné poskytující informaci o aktuálním stavu objektu. Jedná o jednu

z nejčastěji používaných proměnných napříč celým řídicím softwarem i nadřazeným HMI systémem. V řídicím programu zastávají klíčovou úlohu při definování interlocků (blokovacích podmínek), jejichž význam byl popsán výše. Taky jsou hojně využívány při programování řídicích sekvencí a při ovládání jednotlivých technologických prvků. V HMI systémech jsou používány jako vizualizace aktuálních stavů – například status otevřeno signalizován zelenou barvou motorové klapky apod. Obecně pokud má například ventil status otevřeno, znamená to, že ventil má aktivní koncový spínač polohy otevřeno a nemá aktivní koncový spínač polohy zavřeno. Naproti tomu pokud má ventil status otevírání nemá aktivní ani jeden koncový spínač a my jsme schopni usuzovat, že ventil opustil polohu otevřeno, ale ještě nedosáhl polohy zavřeno. Díky těmto informacím jsme schopni programově podchytit každý jednotlivý detail v řízení daného objektu.

Při tvorbě řídicích algoritmů pro sekundární odprášení byly tedy statusy využívány při návrhu téměř každé programové sekvence a především při tvorbě interlocků pro všechna zařízení napříč celou technologií sekundárního odprášení. Obrázek 6.19 potom ukazuje použití statusových proměnných při definování omezujících podmínek pro start oplachového čerpadla.

☐ Network 19 : Flush_Pump_M20_M21



Obrázek 6.19: Segment intelocků s použitím statusových proměnných

Detailnější rozbor použití statusových proměnných a dalších vlastností objektů uvedených v této kapitole bude uveden v kapitolách zabývajících se návrhem řídicích algoritmů.

6.5.2 Blok pro měření analogových veličin AnMeas

Blok pro čtení, zpracování a zobrazení informace z karet analogových vstupů se nazývá AnMeas. Tento blok je v řídicím softwaru použit pro zobrazení napětí a proudů vysokonapěťových sekcí elektrofiltru BETH.

Mezi standartní vlastnosti tohoto bloku patří možnost nastavení limitu pro měřené veličiny – lze definovat čtyři limitní stavy – HiHi, Hi, Lo, LoLo a v návaznosti na překročení dané hladiny generovat varovné hlášení nebo poruchový stav.

HiHi limit je potom nejvyšší možná hodnota, kterou může daný analogový signál reálně dosáhnout. Pokud je překročen tento limit, je generována porucha, jelikož tento stav by neměl nastat. Praktickým příkladem může být teplota, která přesáhne kritickou hodnotu a představuje tak nebezpečí jak pro obsluhu tak technologii samotnou. HiHi limit tedy představuje kritickou hodnotu měřené veličiny, která nesmí být překročena. Hi limit je potom z logiky věci hodnota, která indikuje dosažení vysoké hodnoty měřené analogové veličiny. Pokud je překročen tento limit, je generováno varování, jelikož hodnota je vysoká, ale není ještě kritická.

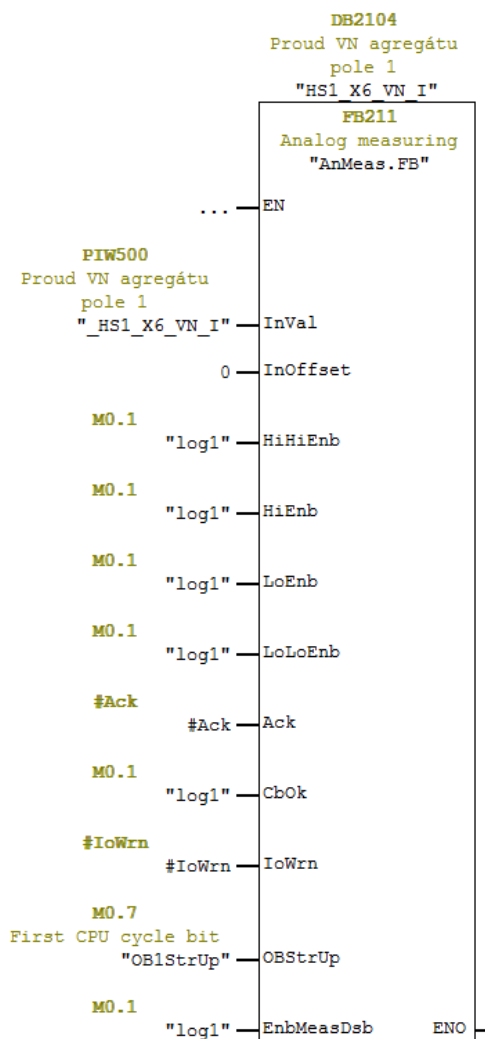
Podobná logika je tedy i v případě Lo a LoLo limitů kdy Lo hodnota představuje nízkou hodnotu, která ale nepředstavuje reálné nebezpečí pro obsluhu a technologii. Zároveň je opět generováno varování. Při dosažení a překročení LoLo limitu, jakožto kriticky nízké hodnoty měřené veličiny je generována porucha. Všechny varování a poruchy jsou pak zobrazeny v nadřazeném HMI systému.

Při praktické realizaci řídicího softwaru je tedy nutné nastavit tyto limity na požadované hladiny tak, aby to odpovídalo požadavkům kladeným na výsledné řízení. To značí změnu všech těchto limitů přes tabulku proměnných (VAT tabulku) jelikož výchozí hodnota těchto limitů je nulová. Pokud nejsou tyto limity nastaveny, nebo jsou nastaveny logicky špatně, jsou generovány varování a alarmy. To vede na nežádoucí stav, jelikož chceme, aby byly varování generovány při přesahu Hi a Lo limitu a poruchy při přesahu HiHi a LoLo limitu.

Address	Symbol	Display format	Status value
DB2104.DBD 202	"HS1_X6_VN_I".HmiInSp.RangeMin	FLOATING_POINT	0.0
DB2104.DBD 206	"HS1_X6_VN_I".HmiInSp.RangeMax	FLOATING_POINT	100.0
DB2104.DBD 210	"HS1_X6_VN_I".HmiInSp.LimHiHi	FLOATING_POINT	90.0
DB2104.DBD 214	"HS1_X6_VN_I".HmiInSp.LimHi	FLOATING_POINT	80.0
DB2104.DBD 222	"HS1_X6_VN_I".HmiInSp.LimLoLo	FLOATING_POINT	10.0
DB2104.DBD 218	"HS1_X6_VN_I".HmiInSp.LimLo	FLOATING_POINT	20.0
DB2105.DBD 202	"HS1_X6_VN_U".HmiInSp.RangeMin	FLOATING_POINT	0.0
DB2105.DBD 206	"HS1_X6_VN_U".HmiInSp.RangeMax	FLOATING_POINT	100.0
DB2105.DBD 210	"HS1_X6_VN_U".HmiInSp.LimHiHi	FLOATING_POINT	90.0
DB2105.DBD 214	"HS1_X6_VN_U".HmiInSp.LimHi	FLOATING_POINT	80.0
DB2105.DBD 222	"HS1_X6_VN_U".HmiInSp.LimLoLo	FLOATING_POINT	10.0
DB2105.DBD 218	"HS1_X6_VN_U".HmiInSp.LimLo	FLOATING_POINT	20.0

Obrázek 6.20: Příklad nastavení hladin analogových hodnot

Konkrétní použití bloku pro zpracování analogových měření je zobrazeno na níže uvedeném obrázku. Jedná se o blok, jenž má pouze vstupy a všechny podstatné parametry nacházející se ve výše rozebrané statické oblasti příslušného instančního DB. Hlavním užitým vstupem pak je InVal, což je vstupní hodnota ve formě analogového signálu z karty. Offset analogového signálu je pak napříč projektem nastaven na nulu, jelikož tyto měřené analogové signály nemají žádný offset. Následují vstupní bity pro povolení jednotlivých hladin HiHi, Hi, Lo, LoLo, na které je nastavena fixní hodnota logické jedničky což reprezentuje, že dané hladiny jsou aktivní. Pokud by vznikla potřeba jednu z hladin vypnout, je možné nastavit danou hladinu na hodnotu logické nuly. Mezi další vstupy patří potvrzovací vstup Ack (Acknowledgement), jež slouží jako zpětná vazba od obsluhy zařízení. Pokud se vyskytne porucha, je generováno poruchové hlášení. Následně po vyřešení poruchy je potvrzeno její převzetí a případná hlášení se zobrazí jako kvitovaná.



Obrázek 6.21: Blok pro realizaci analogových měření AnMeas

Následují diagnostické bity CbOk – kontrola jističe a IoWrn – hardwarová diagnostika vstupní analogové karty. IoWrn jako HW diagnostika je pak totožná pro všechny dále popisované bloky. Doplnkové funkce pak jsou OBStrUp, což je bit sloužící k resetu predikovaného času pro jeden cyklus PLC a EnbMeasDsb což je bit povelu, který umožní přerušování měření analogového signálu.

V konkrétních UDT pak nacházíme další proměnné, které se podílejí na celistvosti zpracování dat tímto blokem. Konkrétně se jedná o UDT pro setpointy, povely, statusy a poruchy. Tyto UDT bloky si pak lze představit jako bloky s uživatelsky definovanými datovými typy, podle nichž lze příslušné datové bloky generovat. Takže pomocí jediného UDT lze generovat nespočet odlišných datových bloků podle potřeby. Jelikož je tento blok použit mnohokrát napříč řídicím programem je existence UDT bloků pro AnMeas velmi výhodná z hlediska efektivity jelikož umožňuje generovat mnoho databloků se stejnou strukturou ale odlišnými hodnotami.

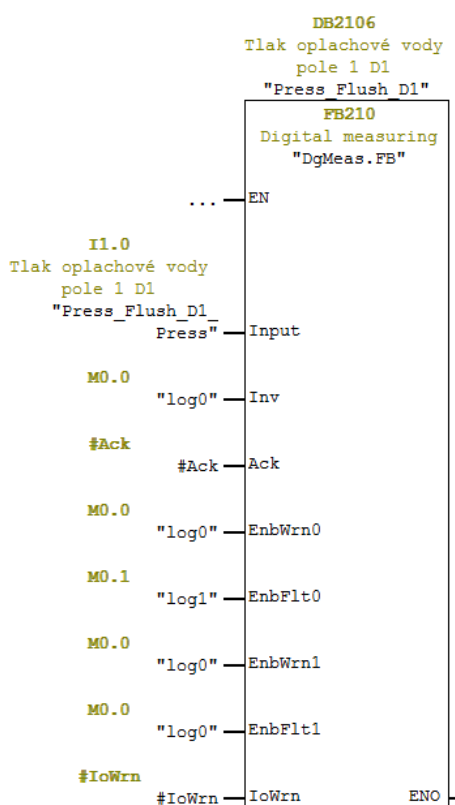
UDT210	AnMeasSp.UDT	STL
UDT211	AnMeasCmd.UDT	STL
UDT212	AnMeasSts.UDT	STL
UDT213	AnMeasFlt.UDT	STL

Obrázek 6.22: UDT bloky funkčního bloku AnMeas

Struktura těchto UDT je pak stejná napříč všemi dále rozebíranými objekty. To značí, že i bloky DgMeas, DevTwoPos mají svá UDT, jež obsahují setpointy, povely, statusy a poruchy. Obsah těchto UDT je potom to, co je odlišné, jelikož se jedná o konkrétní proměnné daného objektu.

6.5.3 Blok pro měření digitálních veličin DgMeas

Blok pro zpracování a zobrazení digitálních signálů se nazývá DgMeas (Digital Measurement). Tento blok získává informace z karet digitálních vstupů. V realizaci softwaru pro řízení sekundárního odprášení je tento blok použit pro zpracování informace od tlakových senzorů pro oplachovou vodu a mlhu. Opět se jedná o blok, jež má pouze vstupní proměnné, přičemž hlavní měřená hodnota je přivedena na vstup bloku pod názvem Input.



Obrázek 6.23: Blok pro realizaci digitálních měření (DgMeas)

Mezi další funkce bloku patří již zmiňovaný Ack vstup sloužící ke kvitaci varovných hlášení a poruch generovaných tímto blokem, dále EnbWrn0 a EnbFlt0, což jsou vstupy, při jejich aktivaci dojde k inicializaci vnitřní logiky, která při detekci změny signálu z logické jedničky na logickou nulu generuje varovné hlášení popřípadě poruchu. Toho lze z výhodou využít ve vizualizaci, jelikož z funkční analýzy je známo, že tlakové snímače vodní mlhy musí být neustále aktivní, jinak dochází ke zhoršení funkce celého odprášení. Proto je tento bit aktivován, přičemž pokud bude například tlakový snímač vlivem poruchy na zařízení neaktivní, bude generována porucha do HMI systému. EnbWrn1 a EnbFlt1 jsou pak nastaveny trvale do nuly, jelikož není nutné generovat varování ani poruchu při detekci logické jedničky neboť tento stav značí, že technologie pracuje správně.

Co se týče struktury UDT je pak totožná z výše uváděnou strukturou UDT bloků AnMeas. To značí, že opět vidíme UDT pro setpointy, povel, statusy a poruchy s konkrétními proměnnými typickými pro objekt digitálního měření.

 UDT200	DgMeasSp.UDT	STL
 UDT201	DgMeasCmd.UDT	STL
 UDT202	DgMeasSts.UDT	STL
 UDT203	DgMeasFlt.UDT	STL

Obrázek 6.24: UDT bloky funkčního bloku DgMeas

6.5.4 Blok pro řízení motorových klap DevTwoPos

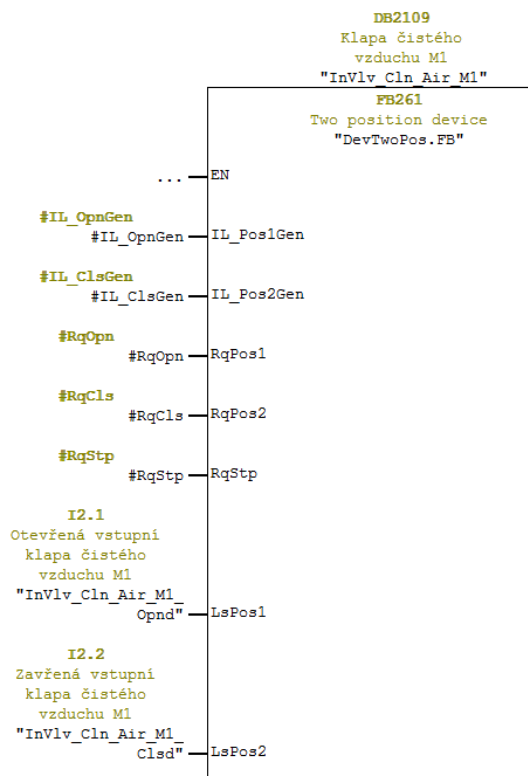
V celém systému sekundárního odprášení se nachází osm motorových klap, přičemž čtyři jsou určeny pro přívod znečištěného vzduchu a další čtyři pak pro odvod již vyčištěného, čistého, vzduchu. Jak víme z předchozí kapitoly, jedná se o zařízení, jež může nabývat dvou fixních stavů, přičemž změnu těchto stavů dosáhneme vhodným řízením příslušné motorové jednotky.

Proto se nabízí použití bloku DevTwoPos (Device Two Position), který slouží k řízení technologických objektů, jež mohou nabývat dvou stavů, přičemž není jasné dáno, který ze stavů je otevřeno a zavřeno. Je proto na programátorovi aby přiřadil pozici 1 a pozici 2 požadovaný význam. V případě tohoto projektu bylo zvoleno pozice 1 jako otevřeno a pozice 2 zavřeno. Dle této logiky byly taktéž napojeny příslušné signály z technologie na vstupy bloku LsPos1 a LsPos2, což jsou vstupy představující koncové snímače poloh (limit swiche) motorových klap.

Celý blok je opatřen vstupy pro blokovací podmínky (interlocky) pro přechod do pozice 1 (v našem případě otevřeno) a do pozice 2 (zavřeno). Tyto podmínky pro přechody do pozic jsou řešeny v samostatných programových částech (networkcích) jež jsou před tímto blokem. Výsledným bitem, jež signalizuje splnění všech blokovacích podmínek je pak InLckPos1 a InLckPos2. Tyto bity jsou pak vstupem do samotného bloku a udávají, zda zařízení může přejít z jedné pozice do druhé.

Na vstupy koncových poloh LsPos1 a LsPos2 jsou přivedeny signály z technologie nesoucí informaci, že daná motorová klapa je otevřená / zavřená. Samotné požadavky (requesty) jsou pak

samostatné signály z technologie, které udávají, že byl vyslán požadavek na otevření / zavření. Ty jsou pak napojeny na vstupy CntrPos1 a CntrPos2.



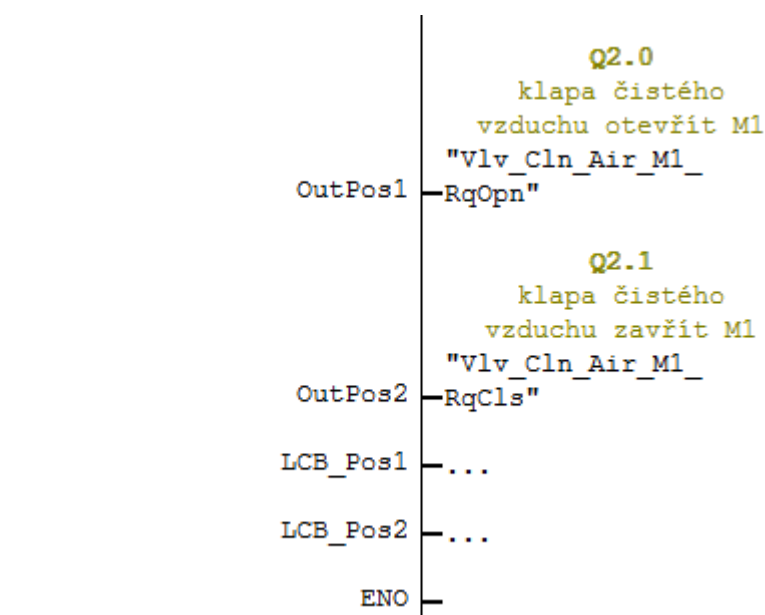
Obrázek 6.25: Blok pro realizaci řízení motorových klap (DevTwoPos)

Mezi další funkce, které nebyly použity v řídicím softwaru, ale je možné je využít patří funkce pro zrušení koncových snímačů (limit switchů) pozice 1 a pozice 2, funkce nastavení trvalé pozice 1 nebo 2, funkce Ack jež je totožná pro všechny bloky.

Taktéž stojí za zmínku režimy, ve kterých může blok pracovat. Existují tři režimy – automaticky (Auto), manuální HMI (ManHmi) a manuální režim řízení pomocí operátorské stanice (ManDesk). V našem případě jsou použity pouze první dva typy režimů. Automaticky režim značí automatickou funkcionalitu bloku v závislosti na řízení pomocí PLC. Jedná se o výchozí stav, ve kterém je blok po většinu času dokud nedojde k podnětu na změnu do manuálního módu s ovládáním z HMI. Manuální mód z HMI pak umožňuje ovládání technologického prvku (v tomto případě motorové klapky) bez limitací, které jsou v řídicím softwaru PLC. To značí, že můžeme klapu zavírat a otevírat libovolně manuálně z nadřazeného HMI systému bez závislosti na PLC programu. Toho lze s výhodou využít při ladění aplikace a uvádění do provozu, kdy již máme definované blokové podmínky, které by nám jinak neumožnili klapu otevřít. Z těchto důvodů je tedy na vstup EnbManHmi (Enable Manual HMI) přivedena logická jednička. Naopak, jelikož nepoužíváme manuální režim ovládání z operátorské stanice, je na vstup EnbManDesk přivedena logická nula.

Dalším použitým vstupem je CbOk na který je přiveden technologický signál informující o vzniku poruchy na dané motorové klapě. Pokud je tento signál aktivní je generována porucha, přičemž je zobrazena v HMI systému. S tím souvisí také HW diagnostika IoWrn totožná s ostatními bloky a detekce přívodu napájení (PsOn).

Co se týče výstupů bloku, jedná se o technologické signály digitálních výstupů, jež informují o vyslání požadavku na řízení do daného technologického objektu (motorových klap). Tato informace je důležitá především pro detekci časových poruch, které byly popsány v kapitole o obecných vlastnostech řídicích prvků. Je-li vyslán požadavek na řízení, je zkoumán čas, za který dojde k fyzickému otevírání dané klap. Podle nastavených časů pro opuštění a dosažení pozice potom dochází k porovnání, zda tento čas nepřesáhl definovaný čas pro opuštění dané pozice.



Obrázek 6.26: Výstupy bloku DevTwoPos

Jedná z nejdůležitějších věcí, která nás zajímá, jsou statusy tohoto bloku, resp. statusy jednotlivých klap, jejichž řízení je realizováno blokem DevTwoPos. Tyto statusy jsou pak v UDT 422 stejné pro všechny objekty typu DevTwoPos. Mezi nejvýznamnější statusy, jež jsou použity v programu pro řízení sekvencí, patří status aktuální pozice (Pos1, Pos2), status přechodu do pozice (MovPos1, MovPos2) nebo status dosažení požadované pozice (Pos1Rchd, Pos2Rchd). Pomocí těchto statusů jsme schopni identifikovat aktuální stav motorové klap a zajistit tak korektní řízení.

UDT420	DevTwoPosSp.UDT
UDT421	DevTwoPosCmd.UDT
UDT422	DevTwoPosSts.UDT
UDT423	DevTwoPosFlt.UDT

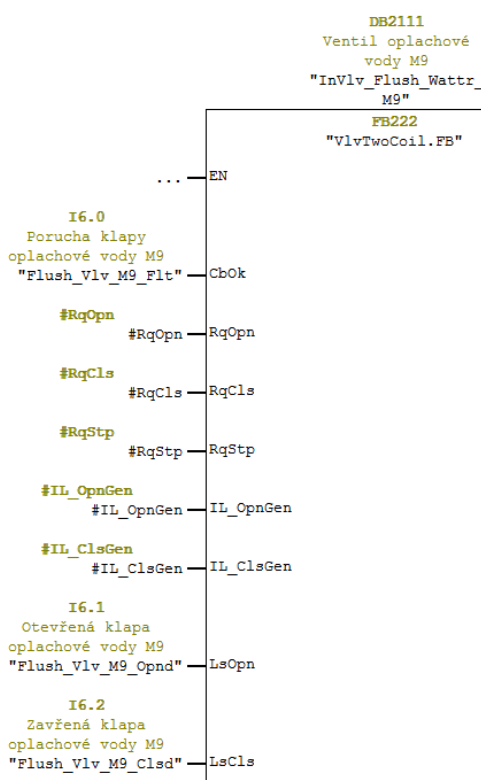
Obrázek 6.27: UDT bloky funkčního bloku DevTwoPos

6.5.5 Blok pro řízení oplachových klap VlvTwoCoil

Kromě motorových klap určených pro přívod špinavého a odvod čistého vzduchu jsou ve stejné technologii taktéž oplachové klapky pro přívod vody z čerpadla, jež jsou aktivní ve fázích oplachu dle dané naprogramované sekvence. Jedná se opět o motorové klapky, nicméně z logiky věci jsou tyto klapky normálně zavřené a přecházejí do otevřeného stavu na krátký časový úsek oplachu pole.

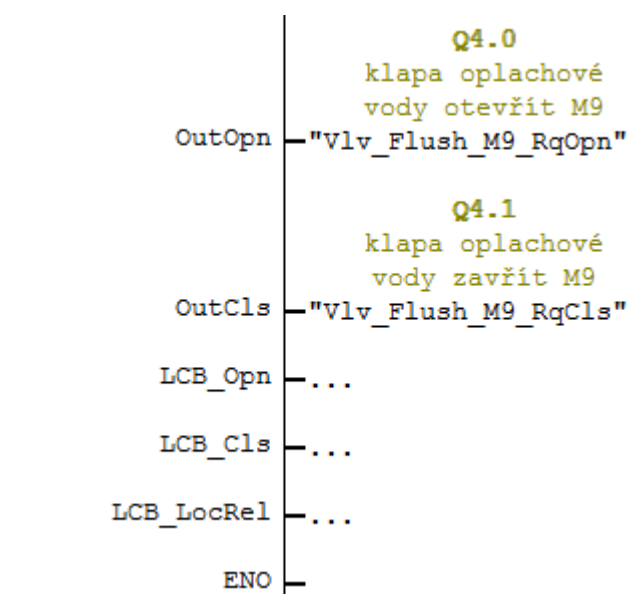
Proto je pro realizaci řízení použit blok VlvTwoCoil (Valve Two Coil), který je standardně použit pro řízení dvou-cívkových ventilů s jasně definovanými polohami otevřeno a zavřeno. Princip dvou-cívkového ventilu spočívá v tom, že při požadavku na otevření se ventil, za předpokladu splnění blokovacích podmínek, začne otvírat a po dosažení koncové polohy setrvá v poloze otevřeno do té doby, než přijde povel na jeho zavření. V tom je hlavní odlišnost od jedno cívkového ventilu, který se po dosažení koncové polohy a konce povelu na otevření začne automaticky vracet do původní polohy. To je důvodem, proč je tento technologický blok opatřen vstupy LsOpn a LsClsd. Pokud ventil dosáhne polohy otevřeno je aktivován signál LsOpn (limit switch open) a naopak pokud ventil dosáhne polohy zavřeno je aktivní signál LsClsd (limit switch closed).

Jelikož se jedná o technologický objekt, který má přímý vliv na stav technologie jsou definovány blokovací podmínky (interlocky) pro přechod do otevřeného/zavřeného stavu a pro zastavení v průběhu otevírání nebo zavírání ventilu.



Obrázek 6.28: Blok pro realizaci řízení oplachových klap (VlvTwoCoil)

Výstupy bloku jsou pak v podobné logice, jako u předchozího bloku pro řízení motorových klap. Jedná se o signály z technologie charakterizující požadavky na otevření a zavření dané oplachové klapy. Na základě této informace víme, že byl vydán povel na otevření (OutOpn) nebo zavření (OutCls) klapy. To, podobně jako v případě předchozího popisovaného bloku, slouží k diagnostikování časových poruch.



Obrázek 6.29: Výstupy bloku VlvTwoCoil

Podobně je to se statusy, které jsou k dispozici v UDT 372 a jež jsou dále použity na nejrůznějších místech v řídicí logice sekundárního odprášení. Jedná se především o statusy signalizující otevření (Opnd), zavření (Clsd), otevírání (Opng) a zavírání (Clsd) jednotlivých klap. Tyto statusy jsou použity taktéž jako blokovací podmínky (interlocky) pro vysokonapěťové sekce, jelikož po dobu oplachu je nutné, aby tyto sekce nebyli aktivní.

Důležité jsou taktéž statusy LSOpn a LSCls, což jsou statusy signálů koncových snímačů poloh otevřeno a zavřeno. Na základě těchto statusů víme, zdali klapa opravdu dosáhla požadované polohy. Co se týče režimů, ve kterých může objekt pracovat, zůstávají stejné jako u bloku DevTwoPos, tedy automatický, manuální (ovládáný z HMI). Režim ovládání z operátorské stanice opět v případě řízení oplachových klap nebyl použit.

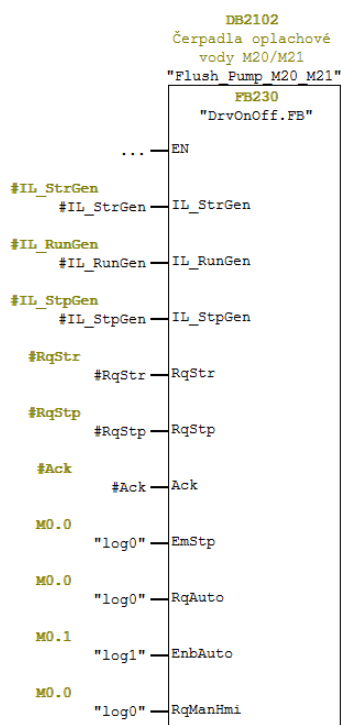
UDT370	VlvSp.UDT	STL
UDT371	VlvCmd.UDT	STL
UDT372	VlvSts.UDT	STL
UDT373	VlvFlt.UDT	STL

Obrázek 6.30: UDT bloky funkčního bloku VlvTwoCoil

6.5.6 Blok pro řízení ventilátorů a čerpadla DrvOnOff

Posledním rozebíraným blokem bude blok sloužící k řízení motorů, přičemž řízený signál nabývá dvou diskretních hodnot nula a jedna. Jinými slovy, motor je spínán a vypínán podle potřeby, nebo podle požadovaných programovatelných sekvencí. Těchto vlastností je možno s výhodou použít při řízení zbylých technologických prvků systému sekundárního odprášení, kterými jsou ofukové ventilátory a čerpadlo dodávající oplachovou vodu.

Samotné řízení těchto prvků je potom realizováno pomocí bloku DrvOnOff, který se zpravidla používá pro neregulované pohony, jejichž řízení spočívá v jejich spínání a vypínání. Opět jsme schopni identifikovat společné prvky s ostatními bloky, jako jsou blokové podmínky (interlocky), kvitovací vstup varování a poruch (Ack), hardwarovou diagnostiku (IoWrn), blokové podmínky (interlocky) nebo volbu režimu ovládání, který je opět automatický nebo manuální z HMI. Výstup bloku je potom pouze jeden, a to požadavek na zapnutí daného technologického objektu. Na tento výstup je pak připojen technologický signál, jež fyzicky spíná ventilátory a oplachové čerpadlo.



Obrázek 6.31: Blok pro realizaci řízení ventilátorů a čerpadla (DrvOnOff)

Mezi nejvýznamnější a nejpoužívanější statusy tohoto bloku potom patří status rozběhu zařízení (Strg), status informující o chodu zařízení (Rung), status zastavování zařízení (Stpg) a status zastavení zařízení (Stpd). S tím souvisejí statusy, jež signalizují, že blokové podmínky pro start (IL_Str), chod (IL_Run) a zastavení (IL_Stp) zařízení jsou splněny. Tyto informace pak lze využít ve HMI systému pro dobrý přehled stavu tohoto technologického prvku.

UDT231	DrvOnOffCmd.UDT
UDT232	DrvOnOffSts.UDT
UDT233	DrvOnOffFlt.UDT

Obrázek 6.32: UDT bloky funkčního bloku DrvOnOff

6.6 Rozdělení řídicích celků technologie

Jak již vyplynulo z tvorby definičního listu, výsledné řídicí algoritmy budou rozděleny do několika logických celků. Jelikož se BETH elektrofiltr skládá se čtyř polí, přičemž každé z těchto polí má své technologické prvky, které je potřeba řídit jsou algoritmy rozděleny do pěti funkčních bloků. Jeden funkční blok BETH_Common je pro technologické prvky, jež jsou společné pro celé odprášení – to jsou ofukové ventilátory, čerpadlo apod. Zbylé čtyři funkční bloky jsou pak pro jednotlivé pole a obsahují algoritmy pro řízení jednotlivých komponent. Zároveň jsou všechny funkční bloky jednotlivých polí umístěny ve funkčním bloku BETH_Common.

FB1100	BETH_Common.FB	FBD	5932
FB1110	BETH_Field_1.FB	FBD	5404
FB1120	BETH_Field_2.FB	FBD	5416
FB1130	BETH_Field_3.FB	FBD	5416
FB1140	BETH_Field_4.FB	FBD	5500

Obrázek 6.33: Funkční bloky obsahující algoritmy pro řízení odprášení TP6

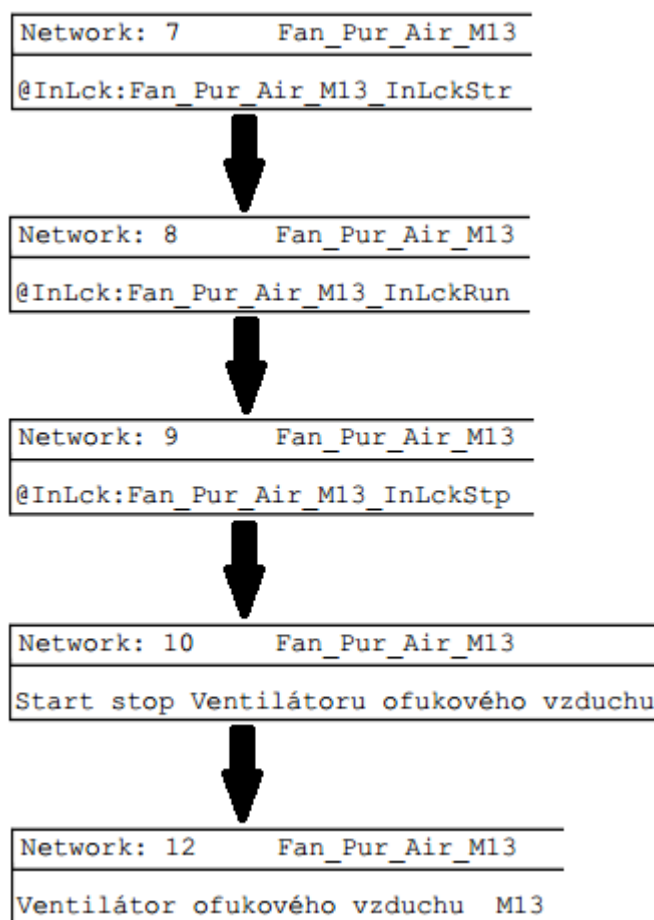
Obsah těchto bloků tedy tvoří hlavní část řídicích algoritmů systému sekundárního odprášení, přičemž jejich detailní rozložení bude popsáno v následujících kapitolách. Obecně lze říci, že všechny tyto funkční bloky mají logickou stavbu, která má svůj význam pro přehlednost kódu. První část každého bloku je tvořena kódem pro obsluhu varování a poruch v HMI (vyčištění struktur poruch a jejich potvrzení) spolu z možností volit módy chodu (automat, manual). Hlavní část je potom tvořena výše rozebíranými bloky, kdy je vždy každému bloku předřazen kód pro obsluhu blokovacích podmínek a kód řídicích algoritmů pro daný technologický prvek. Samotné řídicí algoritmy jsou pak děleny na více částí, dle jejich složitosti tak, aby se zachovala maximální čitelnost programu.

6.6.1 Řídicí algoritmy pro celkové odprášení

Řídicí algoritmy pro celkové odprášení jsou realizovány ve FB1100 pod názvem BETH_Common.FB. Pokud budeme chápat tento funkční blok v jeho hierarchickém rozložení pak začátek kódu je věnován obsluze rutinních procesů, jež musí být vždy vykonány. Mezi tyto operace potom patří kopírování příkazů z HMI stanice, vyčištění struktury poruch, potvrzovací (acknowledge) bit a bit pro nouzové zastavení (Emergency Stop) systému.

Následuje programový kód pro obsluhu jednotlivých technologických objektů. Mezi tyto objekty patří první ofukový ventilátor (Fan_Pur_Air_M13), druhý ofukový ventilátor (Fan_Pur_Air_M14), a oplachová pumpa (Flush_Pump_M20_M21). Každý z těchto objektů má potom nejprve definovány

všechny blokovací podmínky (interlocky) pro start, stop a běh. Dále následuje tzv. start/stop sekvence jež definuje chování daného technologického objektu vzhledem k definované funkční analýze. Hierarchicky poslední je potom samotný funkční blok technologického objektu (DevTwoPos, VlvTwoCoil nebo DrvOnOff) jež zajišťuje samotné řízení objektu v souladu s interlocky a předřazenými programovými sekvencemi.



Obrázek 6.34: Programová struktura návrhu řídicích algoritmů

Stejná programová logika je pak u všech technologických objektů napříč ostatními funkčními bloky ostatních polí filtru. Řídící algoritmy jsou tedy z jedné části tvořeny blokovacími podmínkami, které jsou unikátní pro každý technologický objekt a z druhé části start/stop sekvencemi ve kterých je obsažena samotná logika řízení.

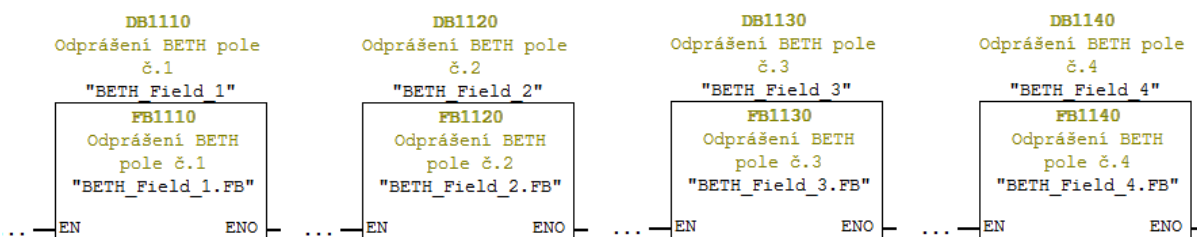
Samotné řídicí algoritmy pro celkové odprášení jsou řešeny dle požadavků funkční analýzy. Řídící logika pro ofukové ventilátorů poté sestává z časovače představující úvodní 30 minutový interval, po jehož uplynutí je aktivován požadavek na zapnutí obou ventilátorů, přičemž tyto ventilátory poběží po celou dobu s výjimkou dojezdu zařízení, kdy je vyslán povel na jejich zastavení. Co se týče blokovacích podmínek pro start ventilátorů, jsou složeny z dílčích interlockových bitů indukujících

otevřenou polohu všech klap čistého a špinavého vzduchu. To zajistí, že je povoleno zapnutí ventilátorů po proběhnutí úvodní sekvence náběhu zařízení. Poslední IL bit je potom určen pro povolení spuštění ventilátorů, pokud jsme v manuálním režimu ovládaném z HMI. Tento bit je vždy reprezentován statusem daného technologického objektu (např. "Fan_Pur_Air_M13".Sts.ManHmi). Použitím tohoto statusu u všech typů IL zajistíme, že zařízení bude plně ovladatelné z vizualizačního systému, což je velmi výhodné při následném ladění programu.

Následuje programový kód pro všechny typy IL a start/stop sekvence pro oplachové čerpadlo. Logický sled je opět totožný s programovou strukturou uvedenou na obrázku 6.34. To značí, že jsou definovány blokové podmínky pro start, zastavení a běh čerpadla s rozdělením na IL pro doběh a IL pro normální provoz a náběh. S tím souvisí taktéž programovatelný čítač polí, který počítá opláchnuté pole tak aby po každém odpichu bylo opláchnuto další pole. Počet polí jako celočíselná proměnná (integer), jež se po každém oplachu navyšuje a tím indikuje oplach následujícího pole. Navíc, dle funkční analýzy, po každém čtvrtém odpichu pece je opláchnut celý filtr (všechny čtyři pole) a čítač je nulován a oplach začíná opět prvním polem.

6.6.2 Řídicí algoritmy pro jednotlivá pole

Řídicí algoritmy pro jednotlivá pole jsou implementovány ve FB 1110, FB1120, FB1130 a FB 1140. Opět pokud budeme vnímat bloky v jejich systematickém rozložení, první část programového kódu slouží pro obsluhu rutinních procesů, jako jsou kopírování příkazů z HMI a čištění struktury poruch. Následují programové bloky pro analogová měření (AnMeas) pro proud a napětí dané vysokonapěťové sekce (dle čísla filtru) a digitální měření (DgMeas) pro indikaci tlaku oplachové vody a vodní mlhy. Jelikož digitální a analogové měření nejsou technologickými objekty s definovanými interlocky není při jejich implementaci dodržovat strukturu programu uvedenou na obrázku 6.34.



Obrázek 6.35: Funkční bloky jednotlivých polí

V případě řízení vysokonapěťových sekcí je zavedena výjimka, jelikož se nejedná o typický technologický objekt, jež by mohl být realizován pomocí některého z definovaných funkčních bloků. Pro ovládání VN sekcí je určen pouze jeden diskretní signál pro zapnutí a vypnutí a jedná zpětná vazba informující o chodu dané VN sekce. Proto je v tomto případě programována pouze start/stop sekvence s tím, že blokové podmínky byly explicitně definovány přímo v naprogramovaných sekvencích.

Následuje programový kód pro řízení obou motorových klap příslušejících vždy k danému poli filtru. To značí, že pokud hovoříme o napěťové sekci HS1 je řešeno řízení motorových klap M1 a M5 jakožto klap, jež zajišťují přívod špinavého (M5) a odvod čistého (M1) vzduchu. Jelikož se již jedná o funkční bloky reprezentující dané technologické objekty, jsou definovány IL pro otevření, zavření a zastavení každé klapu.

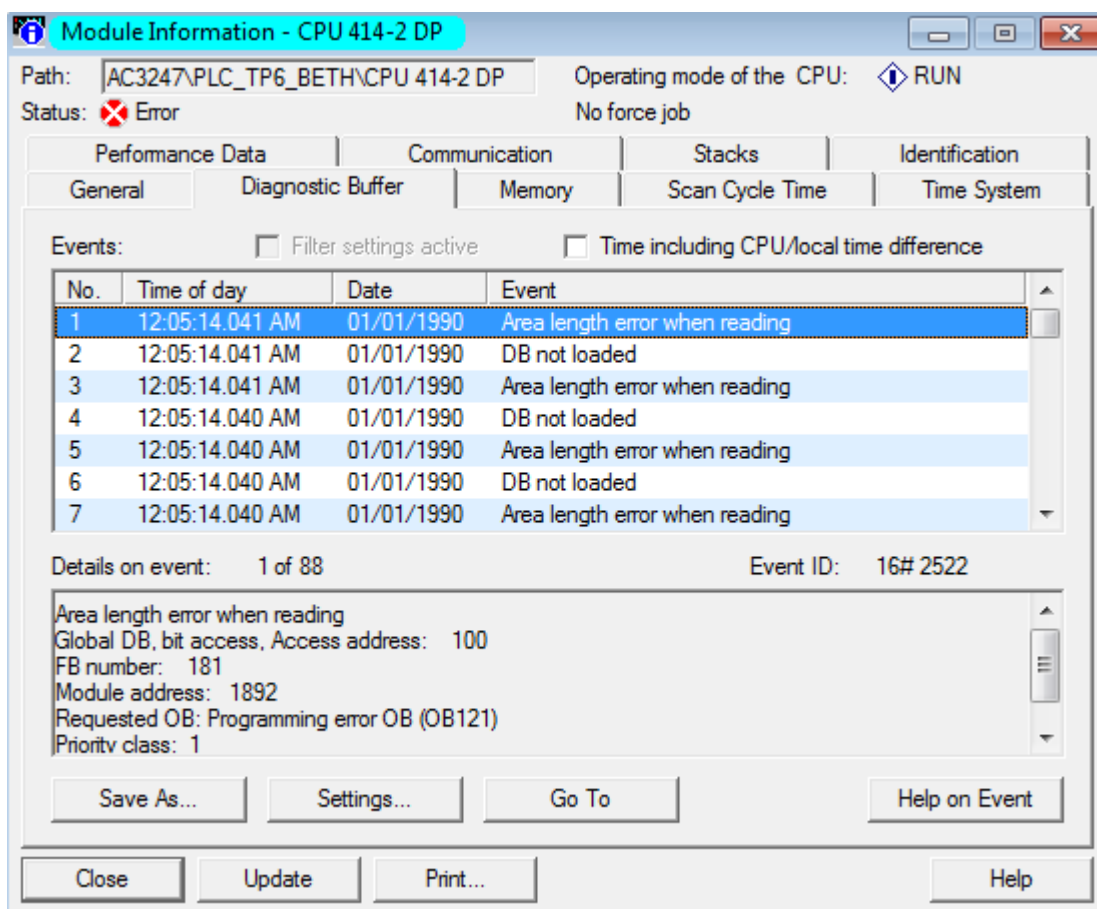
Pro realizaci oplachové sekvence, jež přichází se signálem TP_Ready, je programována start/stop sekvence příslušné oplachové klapu (M9 až M12). Tato sekvence potom přejímá informaci z FB BETH_Common kde je implementován čítač polí a následně je proveden kód, který dle čísla pole otevírá příslušnou oplachovou klapu. V součinnosti s definovanými blokovacími podmínkami je zajištěno, že je vždy opláchnuto korektní pole, přičemž je oplach zahájen požadavkem na proplach (Rq_Proplach). Ten je potom aktivní, jen pokud je zařízení v normálním režimu.

Zde je vhodné uvést obecné ideje aplikované při vývoji těchto algoritmů, jelikož implementace algoritmů dle standardů společnosti Ingeteam a.s. je uvedena v exportovaných dokumentech s kódem, jež jsou součástí elektronické přílohy. První a naprosto stěžejní věc je úhel pohledu, který je nutno zaujmout pokud aplikujeme Ingeteam standart je, že algoritmy jsou tvořeny vždy z pohledu konkrétního technologického objektu. Programátor potom musí uvážit význam prvku v definované funkční analýze a tvořit program tak, aby daný prvek dostal své úlohy, jakou v konkrétním procesu zastává.

Základní myšlenka, která se vine celým návrhem, je globální pohled na probíhající procesy uvnitř sekundárního odprášení. Celé odprášení po nájezdu technologie setrvává v normálním provozu s tím, že současně probíhá výroba tekuté oceli v příslušné tandemové peci (v tomto případě č. 6). Z prostoru haly je odváděn špinavý vzduch, jež slouží jako vstupní veličina sekundárního odprášení. Tento špinavý vzduch je do mokrého elektrofiltru přiváděn pomocí klap špinavého vzduchu (M5 – M8). Aby mohlo čištění vzduchu probíhat je potřeba, aby byla daná vysokonapěťová sekce v provozu a zároveň byli aktivní všechny snímače oplachové mlhy, jelikož mlha zajišťuje korektní funkci filtru. Pokud by tato mlha nebyla přiváděna, zhorší se celkové odprašovací vlastnosti filtru. Pokud tedy neprobíhá oplach, jsou všechny klapu čistého a špinavého vzduchu otevřeny, vysokonapěťové sekce zapnuty, čerpadlo vypnuto, oplachové klapu zavřeny, ventilátory v provozu a je trvale přiváděna vodní mlha. Z hlediska programu je tedy zajištěno, aby vše výše uvedeno bylo splněno, dokud není zaznamenán signál TP_Ready resp. Rq_Proplach značící začátek oplachové fáze. Při oplachové fázi je potom opláchnuto takové pole filtru, které určí čítač pro oplach polí v FB1100. Oplach potom probíhá tím způsobem, že se zavřou příslušné klapu čistého a špinavého vzduchu, vypne se vysokonapěťová sekce daného pole, zapne se čerpadlo a otevře se oplachová klapa. Po uplynutí časového intervalu pro oplach se čerpadlo vypne a oplachová klapa se uzavře. Po uplynutí dalšího časového intervalu se pole vrací do své původní odprašovací funkce a klapu čistého a špinavého vzduchu jsou otevřeny, vysokonapěťová sekce zapnuta. Výjimku poté tvoří sekvence celkového oplachu filtru, která nastává po každém čtvrtku odpichu pece. Tato sekvence je programována totožně z výše popsanou oplachovou sekvencí s tím rozdílem, že jsou vypnuty všechny sekce a zavřeny všechny klapu čistého a špinavého vzduchu.

6.7 Diagnostika systému v prostředí STEP7

Další krokem v logickém sledu navrhování algoritmů je celková diagnostika systému pomocí prostředí STEP7. Pro tyto účely se využívají diagnostické nástroje hardwarové diagnostiky, především pak diagnostický buffer a zprávy generované procesorem (CPU messages). Hardwarová diagnostika nám potom poskytuje detailní přehled o statusu užitých řídicích komponent, stavu využití paměti nebo informace o aktuálním času cyklu.



Obrázek 6.36: Diagnostika navrženého systému – diagnostický buffer

Jak lze vidět z výše uvedeného obrázku, v první fázi ladění programu šlo pozorovat velké množství programových chyb. Tyto chyby byly způsobeny především zápisem do neexistujícího databloku, odkazováním se nebo voláním neexistující programové částí apod. Při vytváření programu pomocí velmi komplexních funkčních bloků popsaných výše, lze s výhodou využít chybové OB určené jak pro synchronní tak asynchronní chyby, jež mohou vzniknout za běhu programu. To nám umožní ladit program za běhu PLC, které by jinak šlo automaticky do STOPu a ladění obecně by nebylo možné, dokud bychom všechny chyby neodstranili. V následující kapitole si tedy popíšeme velmi účinný nástroj, jež je využit především v době vývoje PLC programu.

6.7.1 Využití chybových OB pro diagnostické účely

Pro efektivní zacházení s různými chybami jsou v programovacím prostředí STEP7 vyčleněny speciální organizační bloky. Tyto bloky jsou pak velmi specificky určené pro konkrétní typ chyby, která se může v procesu řízení objevit. Základní členění těchto OB jsou na OB pro zpracování synchronních chyb (OB121, OB122) a na zpracování asynchronních chyb (OB80 – OB87).

Synchronní chyby pak můžeme vnímat jako programovací chyby jako je zápis do neexistujícího DB apod. Důsledkem chyb tohoto typu je především zastavení činnosti PLC (PLC přechází do módu STOP). Použitím bloku OB121 docílíme toho, že PLC setrvá v módu RUN a my jsme schopni s ním pracovat i přes programové chyby. Tento blok chyby neodstraní, pouze zajistí, že nedojde k zastavení chodu PLC. To je výhodné při programování základních technologických funkcí jednotlivých objektů, které zatím nemají naprogramovanou diagnostiku a další doplňkové funkce, které jsou nedílnou součástí všech užitých technologických objektů.

OB80	Cycle Time Fault	FBD
OB81	PS_Flt	FBD
OB82	I/O_Flt1	FBD
OB83	I/O_Flt2	FBD
OB84	CPU_Flt	FBD
OB85	OB_NL_Flt	FBD
OB86	RACK_Flt	FBD
OB87	COMM_Flt	FBD
OB121	PROG_ERR	FBD
OB122	MOD_ERR	FBD

Obrázek 6.37: Užití chybové OB pro diagnostiku řídicího systému

Asynchronními chybami pak rozumíme chyby, jež se mohou objevit náhodně a mají významný vliv na chod řídicího systému. V prostředí STEP7 jsou implementovány specifické organizační bloky pro zacházení s těmito nečekanými událostmi. Na výše uvedeném obrázku pak lze pozorovat aplikaci těchto bloků přímo v řídicím softwaru. Konkrétně jsou tyto OB volány při překročení maximální doby trvání jednoho programového cyklu (OB80), výpadku napájení (OB81), přerušení diagnostiky (OB82), přerušení vodiče na modulu (OB83), hardwarové chybě CPU (OB84), chybovém nahrání volaného OB (OB85), hardwarové chybě lišty na které je umístěno PLC S7-400 (OB86) a při poruše komunikace (OB87).

Implementací těchto OB získáváme komplexní diagnostické „prostředí“ pro celkový řídicí systém sekundárního odprášení. Pokud se tedy objeví porucha kdekoliv v rámci systému je spolehlivě detekována a příslušné chybové hlášení je zobrazeno v nadřazeném HMI systému. Lze tedy hovořit o komplexní hardwarové i softwarové diagnostice zahrnující i diagnostiku komunikace.

6.8 Testování navržených algoritmů pomocí simulace technologie

Testování řídicích algoritmů pomocí simulací navržených v prostředí WinMOD lze vnímat jako před-
fázi konečného uvádění řídicích systému do reálného provozu. Pomocí simulace jsme schopni
simulovat reálné podmínky, jež mohou nastat a odladit software dříve, než technologie najede naostro.
To zajistí vyřešení mnohých chyb, které by se za jinak objevili až při ostrém nájezdu a značně by
prodloužili celý proces uvádění zařízení do provozu. Celý proces tvorby simulačního prostředí pro tuto
technologie je řešen v související diplomové práci [8]. Zde budou pouze uvedeny výsledky testování
v součinnosti s navrženými řídicími algoritmy implementovanými v PLC.

První část testování zahrnuje otestování řízení jednotlivých technologických objektů, které jsou
řízeny pomocí signálů z PLC. Na obrázku 6.36 potom vidíme testování výstupní části technologie –
klap jež odvádějí čistý vzduch ven ze sekundárního odprášení. Testování bylo provedeno pomocí VAT
tabulky, do které byly umístěny statusy jednotlivých klap a následně bylo simulováno jejich otevření /
uzavření.

<div><div>● InVlv_Cln_Air_M1_Opnd</div><div>I2.1</div></div>	<div><div>● InVlv_Cln_Air_M2_Opnd</div><div>I2.5</div></div>	<div><div>● InVlv_Cln_Air_M3_Opnd</div><div>I3.1</div></div>	<div><div>● InVlv_Cln_Air_M4_Opnd</div><div>I3.5</div></div>
<div><div>● InVlv_Cln_Air_M1_Clsd</div><div>I2.2</div></div>	<div><div>● InVlv_Cln_Air_M2_Clsd</div><div>I2.6</div></div>	<div><div>● InVlv_Cln_Air_M3_Clsd</div><div>I3.2</div></div>	<div><div>● InVlv_Cln_Air_M4_Clsd</div><div>I3.6</div></div>

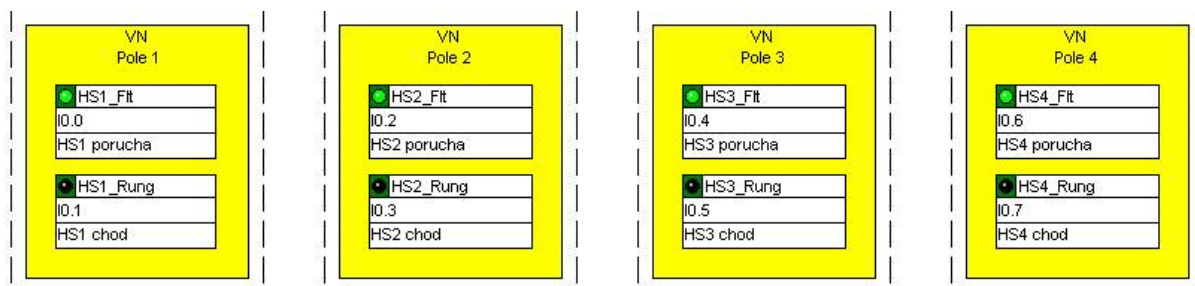
Obrázek 6.38: Simulace řízení výstupních klap [8]

Následně byla otestována funkčnost jednotlivých oplachových klap včetně digitálních senzorů pro
měření tlaku oplachové vody. S tím taktéž souvisí otestování funkce čerpadla oplachové vody a
ofukových ventilátorů. To vše bylo nutné udělat v manuálním režimu nastaveném v nadřazeném HMI
systému jelikož automatický režim má definovány blokovací podmínky dle funkční analýzy a
neumožnil by spuštění daného prvku bez závislosti na ostatních.

<div><div>● Flush_Vlv_M9_Opnd</div><div>I6.1</div></div>	<div><div>● Flush_Vlv_M10_Opnd</div><div>I6.5</div></div>	<div><div>● Flush_Vlv_M11_Opnd</div><div>I7.1</div></div>	<div><div>● Flush_Vlv_M12_Opnd</div><div>I7.5</div></div>
<div><div>● Flush_Vlv_M9_Clsd</div><div>I6.2</div></div>	<div><div>● Flush_Vlv_M10_Clsd</div><div>I6.6</div></div>	<div><div>● Flush_Vlv_M11_Clsd</div><div>I7.2</div></div>	<div><div>● Flush_Vlv_M12_Clsd</div><div>I7.6</div></div>
<div><div>● Press_Flush_D1</div><div>I1.0</div></div>	<div><div>● Press_Flush_D2</div><div>I1.1</div></div>	<div><div>● Press_Flush_D3</div><div>I1.2</div></div>	<div><div>● Press_Flush_D4</div><div>I1.3</div></div>

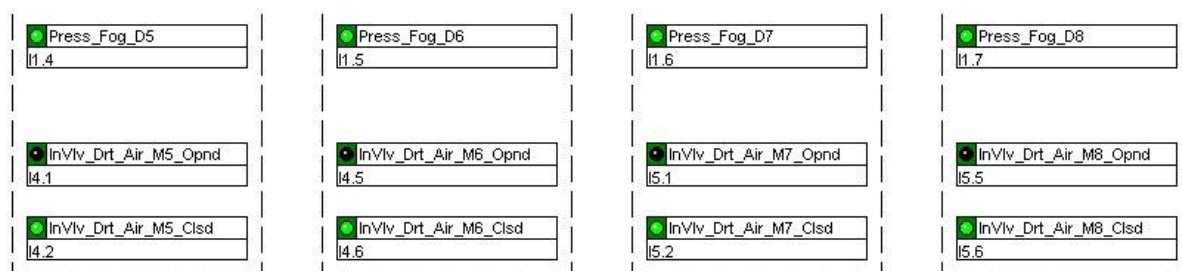
Obrázek 6.39: Simulace řízení oplachových klap [8]

Předmětem dalšího testování bylo otestování funkčnosti vysokonapěťových sekcí elektrofiltru
respektive, jestli je chod každé sekce možný při vyslání příslušného řídicího signálu z PLC. S tím
souvisí taktéž poruchové hlášení unikátní pro každé pole filtru, jež je generováno, pokud
vysokonapěťová sekce nepracuje správně a průmyslový počítač Promethos, jež představuje řídicí
systém elektrofiltru, vysílá chybové hlášení do PLC.



Obrázek 6.40: Simulace řízení vysokonapěťových sekcí elektrofiltru [8]

Poslední testovanou částí je vstupní část technologie tvořena klapami špinavého vzduchu, jež je přiváděn z prostoru haly tandemové pece. Opět byly testovány jejich reakce na povely pro otevření a zavření z PLC. Současně byly otestovány tlakové snímače vodní mlhy, jež by měly být za normálního provozu neustále aktivní, jelikož zajišťují korektní funkci elektrofiltru.



Obrázek 6.41: Simulace řízení vstupních klap [8]

Druhá část testování potom sestává z testování technologie jako celku v souvislosti s definovanou funkční analýzou. Podle této funkční analýzy byly navrženy řídicí algoritmy, jež byly popsány v předchozích kapitolách a které představují automatický režim řízení pomocí PLC. Pokud je řízení v automatu, jsou vykonávány naprogramované sekvence definované v jednotlivých FB.

To značí, že byla testována úvodní sekvence, ke které dochází při nájězdu zařízení. Při této sekvenci dochází k postupnému proplachu všech čtyřech polí elektrofiltru postupně s definovaným časovým intervalem pro každé z polí. Následně jsou zapnuty všechny čtyři vysokonapěťové sekce filtrů a otevřeny všechny klapky čistého a špinavého vzduchu z důvodu lepšího proudění vzduchu do technologie. Při této sekvenci jsou testovány reakce na povely vysílané z PLC a testování doby jež uplyne než klapka opustí výchozí pozici a přejde do pozice druhé. Dále je testováno, jestli je sekvence naprogramována správně a tedy jestli jsou v platnosti všechny definované blokové podmínky a nemůže nastat kritický stav, jež by mohl způsobit fatální následky pro reálnou technologii.

Další testovanou částí je normální stav, ve kterém technologie setrvává po průběhu nájězdu. V případě normálního provozního stavu byla testována oplachová sekvence, jež nastává po každém pátém odpichu tandemové pece. Tento pátý odpich byl simulován signálem TP_Ready jež je přiváděn z technologie a signalizuje možnost zahájení oplachové fáze filtru. Tímto signálem je tedy spuštěna

sekvence, při které se uzavřou klapky čistého a špinavého vzduchu pro dané pole, vypne se vysokonapěťový agregát, spustí se čerpadlo a otevře se příslušná oplachová klapa. Po proběhnutí oplachu se potom klapa uzavře, vypne se čerpadlo, spustí se VN sekce a otevřou se klapky. Opět i zde v normálním provozu jsou testovány reakce jednotlivých klap a ostatních prvků na povely z PLC a testuje se splnění jednotlivých blokovacích podmínek definovaných v programu.



Obrázek 6.42: Simulace chodu celkové technologie sekundárního odprašení [8]

Poslední testovanou částí je doběh zařízení, jež se provádí primárně při odstávce tandemové pece. Jedná se o sekvenci obdobnou nájezdové s drobnými odlišnostmi. Opět jsou propláchnuta všechna pole s daným časovým intervalem a jsou otevřeny všechny klapky pro odvětrání celého systému. Zde je testováno především, zdali všechny komponenty technologie po skončení této sekvence setrvávají v bezpečném stavu, tak aby při případném zásahu obsluhy nebo údržby nedošlo k život ohrožujícímu stavu. To značí, že sekce setrvávají ve vypnutém stavu, ventilátory a čerpadlo jsou vypnuty a všechny klapky jsou otevřeny.

6.9 Uvedení do provozu a testování navržených algoritmů řízení

Konečné uvádění do provozu má vždy několik po sobě jdoucích fází, jež vedou k finálnímu nájezdu technologie do ostrého provozu. První fází, jež se v tomto případě provádí, je celková rekonstrukce tandemové pece při k tomu určené odstávce. Tato rekonstrukce zahrnuje bourání žáruvzdorné vyzdívky uvnitř pece a její znovupostavení. Po této činnosti se může přistoupit k další fázi uvádění technologie do provozu a tou je instalace elektroinstalace, budování kabelových tras a zapojování rozvaděčů. Po těchto krocích se může přistoupit k finálním softwarovým pracím na technologii.

V případě této práce se uvedení do provozu týká testování navrženého řídicího systému uvnitř rozvaděče, jež bude umístěn přímo v technologii sekundárního odprášení. Tyto testy zahrnují především otestování vstupů a výstupů, jež jsou připojeny na relé a zajišťují diskrétní řízení technologie. Dále se testuje korektní komunikace PLC se vzdálenou periferií ET200S a vizualizačním panelem umístěným na dveřích rozvaděče.

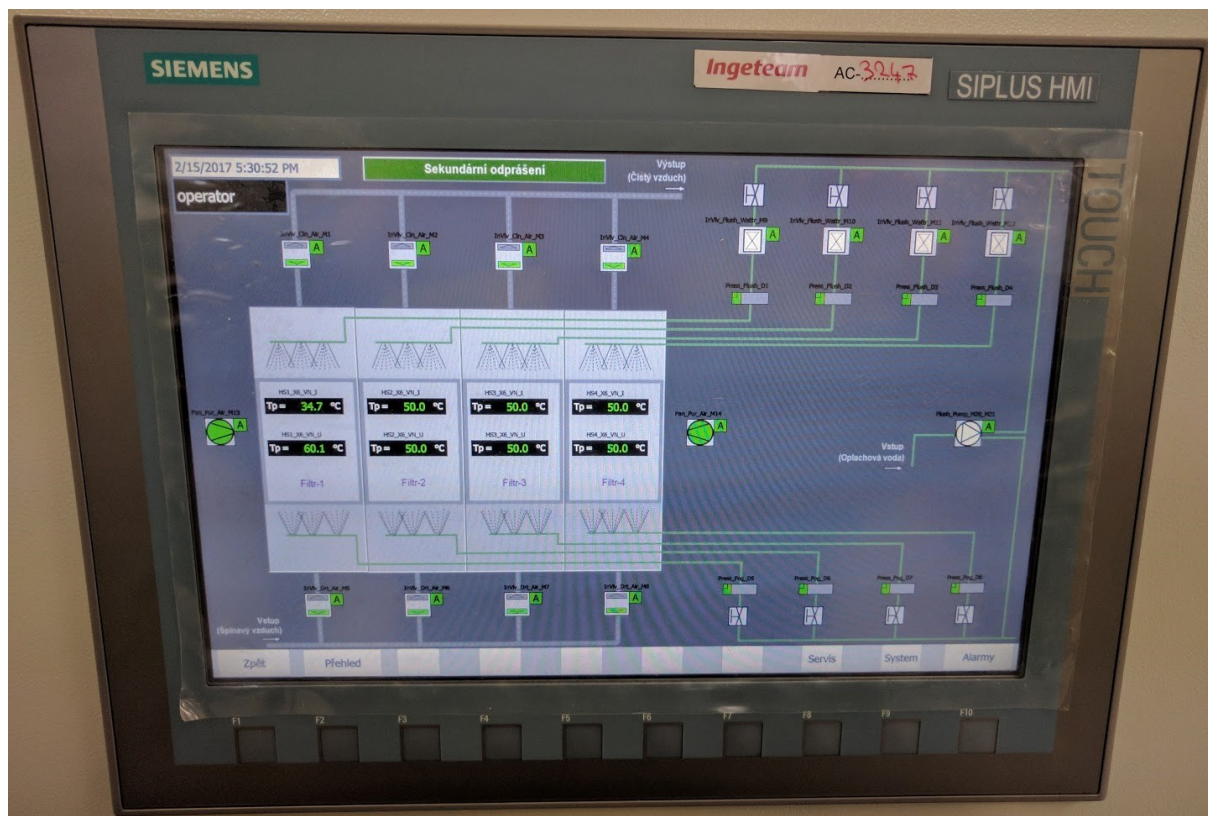
Testování vstupů a výstupů neboli „I/O check“ bylo provedeno jako celek dle navrženého I/O listu a posléze bylo namátkově testováno spínání požadovaných relé odpovídající příslušným digitálním výstupům PLC. PLC Simatic S7-400 bylo propojeno s tlačítkem pro nouzové zastavení na dveřích rozvaděče, s distribuovanou periferií a vizualizačním panelem pomocí PROFIBUSu.



Obrázek 6.43: Výsledný vnitřek rozvaděče určený pro řízení sekundárního odprášení

Po připojení osobního PC do PLC a nahrání navržených řídicích algoritmů bylo taktéž testovat činnost vizualizačního panelu, jehož software byl navržen ve WinCC. Na níže obrázku je možno vidět „online“ stav panelu, který odpovídá stavu před nájездem technologie, kdy klapý jsou uzavřeny a

vysokonapěťové sekce jsou vypnuty. Z testovacích důvodů byly taktéž do DB jednotlivých sekcí nahrány hodnoty napětí a proudu přes VAT tabulku.



Obrázek 6.44: Testování součinnosti vizualizačního panelu a PLC [8]

Výsledek tohoto testování tedy potvrzuje správnou součinnost mezi navrženým řídicím systémem Siemens Simatic S7-400 a všemi dalšími komponenty, které se podílejí na řízení technologie sekundárního odprášení. Otestování HW vstupů a výstupů potom zajišťuje správné vykonávání programových sekvencí, jež jsou definovány uvnitř uživatelského programu v PLC.

7 Závěr

Předmětem této práce byl návrh řídicích algoritmů pro technologii sekundárního odprášení tandemové pece, jež se nachází v Arcelor Mittal Ostrava a.s. Cíl práce, který byl nastíněn v úvodu, byl poskytnout komplexní pohled na řešení projektu v oblasti průmyslové automatizace s respektováním zásad projektového řízení a struktury projektů společnosti Ingeteam a.s.

Návrh řídicích algoritmů předpokládá vždy znalost technologie jako celku a celkový náhled na všechny procesy, jež probíhají uvnitř. V případě sekundárního odprášení se jedná o porozumění jednotlivým sekvencím, které musí být vykonány v určitou dobu a musí být provedeny přesně podle stanovené funkční analýzy. Sestavení korektní funkční analýzy je proto klíčovým krokem při snaze o řízení a implementace algoritmů. V případě sekundárního odprášení je funkční analýza tvořena jednotlivými provozními stavy, jež mohou v průběhu řízení nastat. Konkrétně se pak jedná o nájezd, normální provoz a doběh zařízení, s tím, že každý tento provozní stav má své definované parametry, sekvence a časové konstanty, se kterými pracuje.

Návrh řídicích algoritmů samotných byl poté realizován pomocí programového prostředí pro řešení komplexních automatizačních úloh STEP7 od společnosti Siemens. Jako základ při tvorbě algoritmů byla vytvořena definiční listina všech symbolů v PLC, jež slouží jako základ při generování programových změn pomocí maker. Dále byl vytvořen I/O list neboli seznam fyzických vstupů a výstupů celé technologie, který sloužil jako výchozí materiál při parametrizaci technologických objektů uvnitř řídicího softwaru a následně simulace realizované v prostředí WinMOD.

Implementace řídicích algoritmů poté spočívala v definování struktury automatizačního systému tvořeného programovatelným automatem Siemens Simatic S7-400 a distribuovanou periferií ET200S s komunikačním protokolem PROFIBUS. Hardwarová konfigurace celého systému byla navržena v prostředí STEP7 stejně jako samotný program sestávající primárně z pěti funkčních bloků určených pro řízení samotného odprášení a z doprovodných funkčních bloků pro realizaci diagnostických funkcí systému. Tyto funkční bloky jsou systematicky členěny podle logického uspořádání řídicích celků v definovaném subsystému. Každý funkční blok obsahuje programový kód pro obsluhu rutinních procesů (kopírování příkazů z HMI, čištění struktury poruch apod.) a funkční bloky pro řízení jednotlivých technologických objektů s definovanými blokovacími podmínkami a logikou řízení spolu s diagnostickými funkcemi každého objektu.

Testování navrženého řídicího systému bylo zajištěno pomocí vytvořené simulace v prostředí WinMOD jakožto komplexního softwaru pro realizaci simulací reálných procesů uvnitř řízené technologie. Předmětem testování byly jednotlivé vstupy a výstupy v celé technologii a v návaznosti na to i celé programové sekvence obsažené v navržených řídicích algoritmech. Navržený řídicí systém by posléze testován v rozvaděči, jež bude posléze umístěn v rozvodně sekundárního odprášení. Byla provedena verifikace všech fyzických vstupů a výstupů technologie (I/O check) a testována součinnost řídicího systému s vizualizačním panelem na dveřích rozvaděče. Navržený řídicí systém spolu s řídicími algoritmy lze tedy umístit do provozu za účelem řízení technologie sekundárního odprášení.

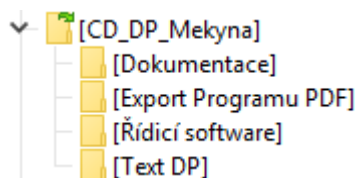
Literatura

- [1] BERGER, Hans. *Automating with SIMATIC S7-400 inside TIA Portal: configuring, programming and testing with STEP 7 Professional*. Weinheim: Publicis Publishing, 2013, 747 s. ISBN 3895783838.
- [2] BERGER, Hans. *Automating with STEP 7 in LAD and FBD: SIMATIC S7-300/400 programmable controllers*. 4th rev. and enl. ed. Weinheim: Wiley-VCH, 2008, 440 s. ISBN 978-3-89578-297-8.
- [3] BERGER, Hans. *Automating with STEP 7 in STL and SCL: programmable controllers SIMATIC S7-300/400*. 4th rev. ed. Erlangen: Publicis Corporate Pub., 2007, 527 s. ISBN 978-389-5782-954.
- [4] BERGER, Hans. *Automating with SIMATIC*. 5th edition. Erlangen, Germany: Publicis Publishing, 2013, 284 p. ISBN 978-3895783876.
- [5] BAŽAN, Jiří a Ladislav SOCHA. *Základy teorie a technologie výroby železa a oceli*. Ostrava: VŠB - Technická univerzita Ostrava, 2014. ISBN 978-80-248-3583-9.
- [6] SIEMENS. *Automation System S7-400 CPU Specifications: Manual*. Germany, 2002.
Dostupné z:
https://cache.industry.siemens.com/dl/files/001/19538001/att_52037/v1/CPU-Daten_e.pdf
- [7] SIEMENS. *ET 200S Distributed I/O System*. Germany, 2004.
Dostupné z:
http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/prumyslove_automatizacni_systemy_simatic/decentralni_periferie/simatic_et_200s/manual_et_200s_12_2004_en.pdf
- [8] GAVLAS, Antonín. *Návrh a realizace vizualizace pro odprášení tandemové pece*. Ostrava: VŠB - Technická univerzita Ostrava, 2017.
- [9] MÁCHA, VIKTOR. *Fotogalerie Arcelor Mittal Ostrava*. Ostrava.
Dostupné z: <https://www.viktormacha.com/galerie/arcelormittal-nova-hut-ostrava-115/>

Seznam příloh

Příloha I: Adresářová struktura přiloženého CD

Struktura přiloženého CD:



Řídicí software – obsahuje řídicí software vytvořený ve STEP7. Projekt obsahuje kompletní řízení technologie sekundárního odprášení dle Ingeteam a.s. standartu.

Export programu v PDF – export funkční bloků obsahující řídicí algoritmy pro sekundární odprášení FB 1100, FB 1110, FB 1120, FB1130 a FB1140 v PDF formátu.

Dokumentace – obsahuje užitou elektro dokumentaci k technologii sekundárního odprášení a jinou dokumentaci týkající se řídicího systému.

Text DP – textová část diplomové práce ve formátu .doc a .pdf.